

**UNIVERZITA KARLOVA V PRAZE**

**Přírodovědecká fakulta**

Katedra aplikované geoinformatiky a kartografie



## **DATOVÉ ZDROJE HODNOCENÍ LAND USE V ČESKU**

### **DATA RESOURCES FOR EVALUATION OF LAND USE IN THE CZECH REPUBLIC**

Bakalářská práce

Jakub Václavík

květen 2014

Vedoucí bakalářské práce: RNDr. Přemysl Štych, Ph.D.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením školitele RNDr. Přemysla Štycha, Ph.D., a že jsem všechny použité materiály řádně citoval.

Jsem si vědom toho, že případné využití výsledků, získaných v této práci, mimo Univerzitu Karlovu v Praze je možné pouze po písemném souhlasu této univerzity.

Svoluji k zapůjčení této práce pro studijní účely a souhlasím s tím, aby byla řádně vedena v evidenci vypůjčovatelů.

V Praze dne 20. 5. 2014

---

Jakub VÁCLAVÍK

**Poděkování:**

Děkuji vedoucímu této práce RNDr. Přemyslu Štychovi, Ph.D. za projevenou trpělivost, věnovaný čas a nespočet cenných rad. Dále děkuji své rodině za podporu v průběhu studia. Také děkuji všem přátelům, kteří mě podporovali, zvláště Kamile Trpišovské, Iloně Svačinové, Martinu Novotnému a Petru Erbenovi.

# Datové zdroje hodnocení land use v Česku

## Abstrakt

Cílem této práce bylo navrhnout a vytvořit nástroj pro záznam chyb v mapách land use a land cover. Byla vyvinuta on-line interaktivní mapová aplikace, která má dva módy. První mód je určen pro vkládání polygonů, které reprezentují chyby v mapě. Druhý mód umožňuje uživatelům hlasovat o validnosti všech polygonů a vyjadřovat tím souhlas nebo nesouhlas s polohovou a tematickou správností polygonu. Uživatelům je také umožněno stahovat jednotlivé polygony ve formátu KML. Hlasování je určeno k ulehčení práce správci aplikace, jedná se o modifikovaný reputační systém, který je používán v dnešním dynamickém webu 2.0.

V práci jsou stručně popsány vybrané datové zdroje land use a land cover. Jedná se o Corine land cover, GlobCover, Urban Atlas a katastrální mapu České republiky. Do aplikace byla pomocí WMS přidána překryvná vrstva katastrálních map. Aplikace byla vytvořena pomocí Google Maps API (Application programming interface), které bylo společně s konkurenčními API popsáno. Aplikace byla vytvořena pomocí skriptovacích jazyků PHP a JavaScript a databázového systému MySQL. Výsledky práce shrnují funkčnost aplikace a poukazují na některé problémy, se kterými se webový vývojář může potýkat.

**Klíčová slova:** webová aplikace, crowdsourcing, land use, land cover, JavaScript, PHP, MySQL

## Data resources for evaluation of land use in the Czech Republic

### Abstract

The main aim of this thesis was design and create tool for recording errors in land use and land cover maps. The on-line interactive map application was developed. The application has two modes. The first mode is designed to inserting polygons, which represent errors in map. The second mode lets user vote on the accuracy of the polygons. Voter expresses agree or disagree with thematic and spatial accuracy. Also, the user can download each polygon KML file. The purpose of voting is to facilitate administrators work. The voting system comes from reputation system, which is used by Web 2.0.

The paper briefly describes the selected land use and land cover data sources. These are Corine land cover, GlobCover, Urban Atlas and cadastral map of Czech Republic. Cadastral layer was added to the application by WMS technologie. The application was created using the Google Maps API, which has been described together with competing API. The application was developed in programming languages PHP and JavaScript and database system MySQL.

**Keywords:** web application, crowdsourcing, land use, land cover, JavaScript, PHP, MySQL

# OBSAH

<b>1 ÚVOD .....</b>	<b>7</b>
<b>2 LITERÁRNÍ REŠERŠE A ÚVOD DO PROBLEMATIKY .....</b>	<b>8</b>
2.1 GEO-WIKI .....	8
2.2 WEB 2.0 .....	9
2.2.1 <i>Mapy.cz API</i> .....	10
2.2.2 <i>Google Map JavaScript API v3</i> .....	11
2.2.3 <i>ArcGIS Web Mapping APIs</i> .....	11
2.3 DATOVÉ ZDROJE LAND USE A LAND COVER .....	12
2.3.1 <i>GlobCover</i> .....	12
2.3.2 <i>Corine land cover</i> .....	13
2.3.3 <i>Urban Atlas</i> .....	14
2.3.4 <i>Katastr nemovitostí České republiky</i> .....	15
2.4 TECHNOLOGICKÉ PROSTŘEDKY .....	15
2.4.1 <i>JavaScript a jQuery</i> .....	15
2.4.2 <i>PHP</i> .....	16
2.4.3 <i>MySQL</i> .....	17
<b>3 METODIKA A REALIZACE APLIKACE .....</b>	<b>19</b>
3.1 NÁVRH APLIKACE .....	19
3.1.1 <i>Logika aplikace</i> .....	19
3.1.2 <i>Grafický výstup a uživatelské rozhraní</i> .....	20
3.1.3 <i>Datové úložiště</i> .....	21
3.1.4 <i>Datové zdroje</i> .....	22
3.1.5 <i>Bezpečnost aplikace</i> .....	23
3.2 ZDROJOVÝ KÓD APLIKACE .....	24
3.2.1 <i>Mapové podklady</i> .....	24
3.2.2 <i>Uživatelské účty</i> .....	26
3.2.3 <i>Vkládání polygonů</i> .....	29
3.2.4 <i>Vykreslování polygonů a stahování KML</i> .....	31
3.2.5 <i>Hlasování</i> .....	33
<b>4 VÝSLEDKY A DISKUZE .....</b>	<b>35</b>

4.1 POPIS VYTVOŘENÉ APLIKACE.....	35
4.2 DISKUSE.....	38
<b>5 ZÁVĚR .....</b>	<b>41</b>
<b>POUŽITÉ ZDROJE .....</b>	<b>43</b>
<b>SEZNAM PŘÍLOH .....</b>	<b>46</b>

# 1 ÚVOD

Hlavním cílem této práce je vytvoření nástroje pro sběr land use a land cover dat. Tento nástroj je vyvíjen jako interaktivní mapová aplikace, která je přístupná všem uživatelům internetu. Dalšími cíli je implementovat vybranou metodu validace dat a popsat a použít vybrané zdroje land use a land cover dat na území České republiky. Dílčím cílem je popsat technologie, které jsou nezbytné pro dynamický mapový web takovým způsobem, aby byl text nápomocen začínajícím webovým vývojářům.

Největší motivací, pro zvolení tohoto tématu, bylo propojení moderních technologií a aktuálního geografického tématu: vývoj a změny v krajině. Toto téma je v současné době hojně diskutované odbornou veřejností jak v zahraničí, tak na půdě Přírodovědecké fakulty UK v Praze, kde se tomuto tématu věnuje projekt LUCC Czechia, což je již šestý projekt na toto téma. V zahraničí je výrazným projektem například Geo-Wiki, který se zabývá sběrem a konsolidací land use a land cover dat. Tento projekt je pro autora značně inspirující a byl jedním z podnětů věnovat se tomuto tématu.

Pro vývoj mapové aplikace jsou použity moderní technologie a přístupy, souhrnně označované jako web 2.0. Tento pojem je chápán jako etapa vývoje internetu, kdy se ze statických stránek stává dynamický prostor, který mohou plnit sami uživatelé. Dochází ke sdílení a kolaboraci myšlenek a informací. Za použití těchto přístupů autor navrhl a vyvinul on-line mapovou aplikaci, do které mohou přispívat všichni uživatelé internetu a validnost dat je alespoň částečně kontrolována reputačním systémem.

Při vývoji aplikace byly použity technologie PHP, JavaScript s knihovnou jQuery a databázový systém MySQL.

## 2 LITERÁRNÍ REŠERŠE A ÚVOD DO PROBLEMATIKY

Pro správné pochopení této práce je nutné definovat několik pojmů, se kterými se setkáváme v celém textu. Základním pojmem této práce je crowdsourcing. Tento pojem je novotvarem, který poprvé použil Jeff Howe v článku pro časopis Wired. Howe (2006) vysvětluje pojem crowdsourcing jako akt, kdy nějaká instituce nebo společnost „outsourcuje“ určitou činnost blíže nedefinované skupině lidí formou otevřené výzvy. Klíčové je použití otevřené výzvy a velký počet potenciálních pracovníků. Od prvního použití slova crowdsourcing se pojem prosadil i v odborných kruzích a je hojně používán také v geografii, zejména v geoinformatice.

Rozvoj moderního crowdsourcingu je autory spojován s nástupem webu 2.0 (HAKLAY, 2010; ALBORS, RAMO, HERVAS, 2008). Crowdsourcing je obvykle používán ve velké skupině lidí, proto bylo obtížné tuto metodu zavést a spravovat před příchodem webu 2.0. V oblasti geografie a geoinformatiky se pro data získaná pomocí metody crowdsourcing užívá slovní spojení volunteered geographic information (VGI), které lze volně přeložit jako geografické informace získané od dobrovolníků (GOODCHILD, LI, 2012). Jeden z nejznámějších VGI projektů OpenStreetMap (OSM), který si klade za cíl vytvořit bezplatnou celosvětovou digitální mapu, popisuje a hodnotí Haklay (2010). Dalším webovým mapovým projektem, který je založen na crowdsourcingu, je Geo-wiki. Tento projekt je vyvíjen skupinou, kterou vede Fritz (2009). Protože projekt Geo-wiki je hlavním zdrojem inspirace pro tuto práci, považuji za vhodné jej podrobněji rozebrat v následující kapitole.

### 2.1 Geo-Wiki

Projekt Geo-Wiki je celosvětová síť dobrovolníků, jejichž cílem je sbírat land cover data a zlepšovat současné land cover mapy. Dobrovolníci jsou prostřednictvím webové aplikace dotazováni na takzvané „hotspot“, což mohou být například oblasti, kde jsou různé land cover mapy v nesouladu. Dobrovolník využije buď aplikaci Google Earth, nebo vlastní znalost oblasti a rozhodne, jestli jsou land cover data chybná, případně o jaký typ chyby jde. Odpověď je zaznamenána do databáze společně s fotografií konkrétní oblasti a později využita k opravě stávajících land cover map, nebo vytvoření nové. S vzrůstající kvalitou obrazových dat v Google Earth a jednoduchostí aplikace Geo-wiki jsou nároky na dobrovolníky nižší a přispívát do projektu může stále širší veřejnost (FRITZ, 2009).



V rámci projektu Geo-Wiki je pro získání co největšího objemu dat s co nejobjektivnější vypovídací hodnotou využíváno několik datových zdrojů. Land cover data reprezentují mapy GlobCover, MODIS a GLC2000. Pro vizuální kontrolu je použit projekt Panoramio, který obsahuje geolokalizované a orientované fotografie a Google Earth (GEO-WIKI, 2014).

Autoři vyzdvihují dva hlavní problémy. Prvním problémem je získat více dobrovolníků, kteří by se účastnili projektu Geo-wiki. Nabízí se oslovit studenty univerzit a středních škol, nebo činnost zatraktivnit pomocí internetových her, ve kterých by byla validace land cover dat zakomponována. Druhým problémem je garance určité míry kvality a zabránění zneužívání projektu. Pro věrohodnost projektů, které jsou postavené na crowdsourcing systému hovoří srovnání encyklopedií Wikipedia a Britannica, které ukázalo, že encyklopedie nemají mezi sebou velký rozdíl (GILES, 2006).

Hlavním cílem projektu Geo-wiki je vytvořit celosvětovou hybridní land cover mapu, která konsoliduje různé stávající land cover mapy a data získaná pomocí nástroje Geo-wiki pomocí geostatistických metod (FRITZ, 2009).

## 2.2 Web 2.0

Na poli geoinformatiky a geoinformačních systémů je etablováno několik softwarových společností, které poskytují mnoho komplexních produktů. Většina úsilí a vývoje se zaměřovala na desktop produkty, které kromě svých kvalit a výhod mají množství nevýhod. Často jde o finanční a hardwarové nároky, komplikovanost, nízkou uživatelskou přívětivost a intuitivnost prostředí a často jsou použity zbytečně robustní systémy, které daleko převyšují nároky uživatele. To vše vede k tomu, že geoinformační systémy jsou úzce vymezeny jen pro skupinu odborníků a laická veřejnost není schopna využívat velký potenciál geoinformačních systémů. Situace se začala dramaticky měnit s nástupem webových technologií a přístupů, souhrnně označovaných jako web 2.0. Hlavní myšlenkou je změna internetu ze statické a jednoduché prohlížečky obsahu na dynamickou a interaktivní platformu, která může snadno nahradit desktop aplikace. Obsah již není statický, ale internet je prostorem, který mohou plnit sami uživatelé. Dochází zde ke sdílení a interakci mnoha myšlenek, které již nemají centrální zdroj a správu. Bez centrální správy obsahu informací je nutné zavádět reputační systémy, které filtrují obsah místo redaktorů (ZBIEJCZUK, 2007). Prosazují se zejména technologie, které umožňují spolupráci skriptů na straně klienta a na straně serveru v asynchronním režimu. Při odesílání požadavků na server a zpětném přijímání dat pak není třeba znovu načítat stránku. Tím je dosaženo požadované interaktivity a plynulosti práce, která se blíží desktop aplikaci. Webové aplikace mají proti desktop aplikacím zásadní výhodu v distribuci. Není potřeba je instalovat, aktualizovat a spravovat na každém počítači zvlášť, protože jejich jádro a aplikační logika je na jednom nebo omezeném počtu serverů (ARDIELLI, 2009).

V průběhu posledních let některé velké společnosti působící v oblasti informatiky, například Google, Microsoft nebo Seznam, pronikají do oblasti webových technologií, které lze využít v geoinformatice. Tyto společnosti nabízejí přístup k velkému množství prostorových dat a nabízejí také své API (application programming interface). Rozhraní pro programování aplikací (API) umožňuje ostatním programátorům využívat funkcionalitu velkých webových mapových aplikací, jako je Google Maps, a kombinovat ji s jinými zdroji a vyvíjet zcela nezávislá a individuální řešení. Tyto nové aplikace, které využívají více zdrojů dat, se nazývají mashup. Mashup aplikace jsou velkým přínosem pro poskytovatele API, na kterých jsou aplikace postavené. Softwarové firmy mohou sledovat, jaké funkce uživatelům chybí, pružně na to reagovat a vyvíjet vlastní řešení. Na druhou stranu jsou mashup aplikace vhodné pro úzce specializované nebo nekomerční projekty, které nemají prostředky pro vývoj celých vlastních aplikací (ARDIELLI, 2009).

V následující kapitole je stručný přehled a popis API některých významných mapových aplikací, které budou brány v potaz při výběru vhodné technologie pro tvorbu výsledné aplikace.

### 2.2.1 Mapy.cz API

Mapy.cz je významným mapovým projektem na českém internetu. Základní mapa obsahuje zjednodušené zobrazení dopravní sítě a pozemních staveb, zobrazuje základní typy porostu a interaktivní body zájmu. V rámci ČR existuje pro každou úroveň přiblížení samostatný mapový podklad, tím je optimalizováno množství zobrazovaných prvků a zvýšena čitelnost a přehlednost mapy. Aktualizace mapových podkladů probíhá přibližně každé dva až tři dny a realizuje ji kartografická redakce, kterou lze v případě nalezení chyby kontaktovat. Mezi základní funkce mapy patří ovládací prvky, odečítání souřadnic v systému WGS 84, měření vzdáleností, přepínání mapových vrstev a plánovač tras (SEZNAM, 2014).

Mapy.cz API poskytují množství funkcí pro práci s mapovým polem, mapovými ovládacími prvky, operace se souřadnicemi, překryvnými vrstvami, značkami, vektorovými prvky, dopředným a zpětným geokódováním atd. S příchodem verze Mapy API 4.0 – Marco Polo došlo k celé řadě vylepšení a optimalizací proti verzi 3.0. V sérii verzí 4 se jedná především o následující změny: nový způsob připojení API do stránek, umožňující i asynchronní načítání, podpora Internet Explorer 9, Simple API (pro nejjednodušší užití API) a další (SEZNAM, 2014).

V současnosti je dle licenčních podmínek používání Mapy API zcela zdarma i pro komerční účely a mezi nabízené funkce patří jak zobrazování mapových podkladů a uživatelské rozhraní pro práci s podklady tak vyhledávání v databázi adres a míst (SEZNAM, 2014).

### 2.2.2 Google Map JavaScript API v3

V roce 2005 spustila společnost Google aplikaci Google maps, kterou v předešlém roce odkoupila od firmy Where 2 Technologies. Aplikace je založená na zobrazení, které je odvozeno od Mercatorova zobrazení a souřadnice jsou udávány v systému WGS 84. Základní aplikace poskytuje velmi podobnou funkcionalitu, jaká byla popsána u Mapy.cz. Topografické podklady jsou obecně na území České republiky méně podrobné a méně aktuální než tomu je u konkurenční aplikace společnosti Seznam.cz.

Všechny aplikace užívající Google Map JavaScript API v3 musí mít v hlavičce dokumentu jedinečný identifikační klíč, který je generovaný v rámci Google účtu uživatele. Toto opatření umožňuje společnosti Google kontaktovat uživatele v případě potřeby, například překročení limitů. Aplikace mohou být provozovány zdarma, pokud nepřekročí limit 25 000 přístupů za den po dobu devadesáti po sobě jdoucích dnů. Takto benevolentní licenční ujednání dovolují provozovat i poměrně vytěžované aplikace zdarma. Na druhou stranu licenční ujednání nedovolují používat API v rámci intranetu nebo v síti chráněné firewallem.

API poskytují základní funkce, shodné s popsanými v kapitole Mapy.cz API. Mimo to poskytují mnoho dalších tříd a metod. Mezi nejzajímavější patří možnost stylovat a měnit barevné schéma samotné mapy nebo knihovna pro interaktivní zakreslování polylinií, polygonů a jednoduchých obrazců (GOOGLE, 2013).

### 2.2.3 ArcGIS Web Mapping APIs

Společnost ArcGis poskytuje webové mapové API pro tři různé skriptovací jazyky:

- ArcGIS API for Flex
- ArcGIS API for Silverlight
- ArcGIS API for JavaScript

V následující části bude popsán ArcGIS API for JavaScript, protože výsledkem této práce bude webová aplikace ve skriptovacím jazyce JavaScript, a také proto, že ostatní výše uvedené API jsou v jazyce JavaScript.

#### ArcGIS Web Mapping API for JavaScript

V rámci užívání API a dalších služeb poskytuje společnost ESRI uživatelům nepřehledné množství mapových podkladů. Uživatel tak může získat družicové snímky, topografické mapy, batymetrické mapy, demografické mapy, mapy rizik a ohrožení, mapy land cover, mapy druhového výskytu a mnoho dalších.

API lze používat dvojím způsobem. Uživatel může odkazovat svoji aplikaci na API, které hostuje společnost ESRI na svých serverech, nebo lze celé API stáhnout společně s dokumentací. Toto řešení může být vhodné, když uživatel nemá trvalý nebo žádný přístup k internetu. V obou případech jsou API poskytovány zcela zdarma. Možnosti ArcGIS API jsou obrovské a daleko převyšují konkurenční API, která byla zmíněná výše. Detailní a komplexní dokumentace popisuje všechny obsažené třídy včetně jejich konstruktorů, metod, vlastností a událostí (ESRI, 2014).

API for JavaScript je optimalizováno a částečně vyvíjeno v JavaScript framework Dojo Toolkit. Tento framework výrazně šetří vývojářské kapacity, snižuje počet chyb v kódu a podporuje bezpečnost aplikací.

Kromě kvalitní dokumentace disponuje ArcGIS širokou uživatelskou základnou. ESRI hostuje a moderuje on-line fórum týkající se API for JavaScript, které obsahuje bez mála 7 000 diskusních témat, také poskytuje on-line kurzy, pravidelně vydává zprávy o vývoji a užívání API a dává k dispozici mnoho hotových řešení ve formě kódu.

## **2.3 Datové zdroje land use a land cover**

V následujících podkapitolách bude stručný přehled a charakteristika vybraných datových zdrojů land cover a land use, které jsou dostupné pro území ČR. Při vývoji mapové aplikace budou brány v potaz a některé z nich budou do aplikace zahrnuty.

### **2.3.1 GlobCover**

V roce 2004 byl pod záštitou Evropské vesmírné agentury (European Space Agency, ESA) spuštěn projekt GLOBCOVER. Postupem času se zapojovaly další organizace a na projektu později pracovaly ESA, FAO, UNEP, JRC, IGBP a GOFC-GOLD. Cílem bylo, mimo jiné, vytvořit globální land-cover mapu, která by nahradila a kvalitativně předčila dosavadní projekty, zejména GLC2000. Data byla získána multispektrálním senzorem MERIS (Medium Resolution Imaging Spectrometer), jehož nosičem byla družice Envisat v období od prosince 2004 do června 2006. Interpretací družicových snímků vznikla mapa s prostorovým rozlišením 300 m a klasifikační legendou založenou na LCCS (Land Cover Classification System of UN). Klasifikační legenda je hierarchicky rozdělena a má dvě úrovně. První úroveň je nazývána globální legenda a druhá úroveň regionální legenda. Globcover land cover mapa byla vytvořena jako globální a konzistentní, proto její globální legenda byla navržena tak, aby jednotlivé kategorie byly rozpoznatelné a porovnatelné po celém světě. Globální legenda je rozdělena do 22 základních tříd jejichž výčet je v příloze 1 této práce. Legenda je plně kompatibilní s GLC2000 klasifikací (ESA, 2008). Regionální legenda je tematicky přesnější a zachycuje informace na regionální úrovni. Patnáct kategorií globální úrovně je rozděleno do 41 kategorií úrovně regionální.

Pro Globcover land cover mapu byla použita cylindrická ekvidistantní projekce Plate Carree a souřadnicová soustava WGS 84 (ESA, 2008).

V roce 2010 byl projekt Globcover provozován společně agenturou ESA a Katolickou univerzitou v Lovani. V tomto roce byla vytvořena série produktů odvozená z dat získaných senzorem MERIS v roce 2009. Jedním z produktů byla mapa Globcover land cover, jejíž klasifikační legenda a další vlastnosti jsou velmi podobné předešlé mapě z roku 2005 (ESA, 2011).

Mapy Globcover 2005 ve verzi 2.2 a Globcover 2009 ve verzi 2.3 jsou poskytovány jako binární soubory ve formátu GeoTIFF nebo ve formátu lyr. Mapy jsou distribuovány jako mozaiky celého světa s globální legendou nebo jako samostatné mozaiky pro jednotlivé regiony s regionální legendou. V současnosti ESA neposkytuje žádná data získaná v rámci projektu Globcover prostřednictvím Web Map Service.

### 2.3.2 Corine land cover

Coordination of Information on the Environment (CORINE) je program Evropské komise, spuštěný v roce 1985 a zaměřený na shromažďování informací, týkajících se životního prostředí. Jednou z částí programu je CORINE land cover projekt, který má za cíl poskytovat konzistentní geografická data původně na území dvanácti členských států (EEA, 1995). V roce 1991 rozhodla Evropská komise rozšířit program CORINE i na státy střední a východní Evropy, tím byla do programu začleněna Česká Republika. Mezi roky 1993 a 1996 byla zpracovávána vektorová databáze, která je tvořena interpretací družicových snímků z nosiče LANDSAT 5, které byly nasnímány mezi roky 1986 a 1995 senzorem Multispectral Scanner s prostorovým rozlišením 50 m. Výstupem byla mapa vegetačního pokryvu v měřítku 1 : 1 000 000 s hierarchickou klasifikační legendou, rozdělenou do tří úrovní. Vznikla tak databáze CORINE land cover 90 (CLC90). Nejdetailnější úroveň obsahovala 44 tříd, z nichž se na území ČR vyskytuje 28. V příloze č. 2 této práce je tabulka všech tříd. Nejmenší mapovací jednotkou v projektu CLC byly souvislé plochy o velikosti 25 ha a šířce 100 m. Očekávaná tematická přesnost byla 85 %, nicméně tato přesnost nebyla naplněna (COPERNICUS, 2013).

Po vzniku Evropské agentury životního prostředí (European Environment Agency, EEA) jí byla předána odpovědnost za program CORINE včetně dílčího projektu CORINE land cover. Na evropské i národní úrovni byla vyjádřena potřeba aktualizace a v roce 1999 byly započaty projekty IMAGE2000 a CLC2000. Na projektech pracovala EEA společně s JRC (Joint Research Center) Evropské komise. Projekt IMAGE2000 představoval databázi družicových snímků pořízených z nosiče LANDSAT 7 radiometrem Enhanced Thematic Mapper Plus s prostorovým rozlišením 25 m. Tyto snímky byly zdrojem dat pro aktualizaci projektu CLC a vytvoření databáze změn mezi roky 1996 a 2000. Nejmenší mapovací jednotka pro změnové databáze byla stanovena na 5 ha. V rámci CLC2000 byla také nutná korektura databáze CLC90, protože v ní bylo nalezeno mnoho chyb. Databáze pro rok 2000 má shodnou klasifikační legendu a minimální mapovací jednotku jako

databáze CLC90. Na rozdíl od CLC90 byla splněna očekávaná tematická přesnost 85 % (BÜTTNER, 2002).

V roce 2006 získal odpovědnost za CORINE Land Cover projekt Global Monitoring for Environment and Security (GMES), který byl později změněn na projekt Copernicus. Ve spolupráci GMES a Ministerstva životního prostředí došlo k aktualizaci databáze CLC k referenčnímu roku 2006 a vytvoření změnové databáze mezi roky 2000 a 2006. Data byla získána pomocí družic SPOT 4 a 5 a IRS-P6. CLC2006 přebírá klasifikační legendu z předešlých projektů. V rámci projektu Copernicus je nyní vytvářena databáze CLC2012 (COPERNICUS, 2013).

Databáze CLC90, CLC2000, CLC2006 a změnové databáze jsou dostupné v on-line prohlížeči dat na webových stránkách projektu Copernicus. Lze je také stáhnout z webových stránek EEA v binárním souboru ve formátu tiff nebo shp.

### 2.3.3 Urban Atlas

Ačkoli je pro Evropská města k dispozici široká škála socio-ekonomických dat, z výsledků projektu Urban Audit vyplynulo, že v oblasti land use se města potýkají s nedostatkem dat. V rámci nápravy tohoto nedostatku vznikl projekt Urban Atlas. Urban atlas je vytvářen pod záštitou Land Monitoring Core Service evropského programu Copernicus. V současné době poskytuje detailní a srovnatelná land use data pro 305 evropských měst, včetně jejich zázemí, za rok 2006. Na území ČR pokrývá Urban Atlas 13 měst s širokým zázemím. Celkem je pokryta přibližně třetina území (EEA, 2010).

Atlas byl vyhotoven v měřítku 1 : 10 000. Nejmenší mapovací jednotka je 0,25 ha pro klasifikační třídu 1 a 1 ha pro klasifikační třídy 2 až 5. Výjimkou je případ, když je plocha rozdělena dopravní sítí do více ploch, pak jsou mapovány všechny dílčí plochy až do minimální rozlohy 500 m<sup>2</sup>. Omezena je také minimální mapovací šířka liniových objektů na 10 m. Mapovány jsou také zúžení liniových objektů pod 10 m, pokud je jejich zúžení kratší než 50 m. Klasifikační legenda je založena na nomenklatuře Corine land cover a je hierarchicky rozdělena do tří úrovní. Nejvyšší úroveň obsahuje čtyři třídy, z nichž je dále dělena pouze třída umělé povrchy. Celkem je rozlišeno 22 tříd. Kompletní nomenklatura je umístěna v příloze č. 3 této práce. Minimální tematická přesnost je 85 % pro umělé povrchy a 80 % pro ostatní. Pro tvorbu byly využity národní socioekonomické statistiky, stávající topografické a land use mapy a především snímky z družice Spot 5 s prostorovým rozlišením 2,5 m a z družic s obdobným rozlišením, např. QUICK BIRD či RAPIDEYE. Aktualizace atlasu je plánována na každé tři roky (COPERNICUS, 2011).

Pro prohlížení dat je k dispozici on-line webová aplikace a pro komplexnější analýzy a aplikace lze data stáhnout v binárním souboru ve formátu shapefile pro každé město samostatně. Data jsou také distribuována formou WMS.



### 2.3.4 Katastr nemovitostí České republiky

Katastr nemovitostí je veřejný soubor údajů o nemovitostech České republiky. Vznikl na základě zákona 344/1992 o katastru nemovitostí České republiky. Zahrnuje popis, soupis a polohové určení evidovaných nemovitostí. Z definice katastru nemovitostí vyplývá, že mimo jiné obsahuje údaje o využití krajiny, ve formě druhu pozemku a způsobu jejich využití. Povinnost evidovat druh pozemku a způsob jejich využití je zakotvena ve vyhlášce č. 357/2013 Sb. ze dne 1. 11. 2013 v odstavci 10. Dle přílohy zákona rozlišuje katastr nemovitostí deset druhů pozemků a 29 způsobů jejich využití. Z toho je v katastrální mapě rozlišeno patnáct druhů pozemků a jejich využití za pomoci mapových značek katastrální mapy (ČESKÁ REPUBLIKA, 2013). Mapa vzniká buďto přímým měřením geodetickými metodami, nebo digitalizací starších analogových map a je v systému S-JTSK a vztahném měřítku 1 : 1000. Geometrická aktualizace stávajících pozemků musí probíhat prostřednictvím geometrického plánu, který je vyhotoven na základě zeměměřických prací. Tematickou aktualizaci, respektive změnu druhu pozemku, nebo jeho využívání je vlastník pozemku povinen ohlašovat. Polohová přesnost je v celé katastrální mapě značně heterogenní. Je to způsobeno tím, že některé pozemky jsou převzaty z map stabilního katastru, které dosahovaly chyb až v řádech metrů, a jiné jsou na základě geometrických plánů přesně zaměřeny. Katastrální mapa je přístupná prostřednictvím webové aplikace nahlížení do katastru nemovitostí, nebo pomocí WMS standardu ve verzi 1.1.1, který spravuje Open Geospatial Consortium. Oba přístupy poskytuje Český úřad zeměměřický a katastrální (ČÚZK, 2013).

## 2.4 Technologické prostředky

Následující oddíl pojednává obecně o použitých technologiích a předpokládá, že čtenář disponuje základními znalostmi v oblasti webového vývoje. Budou popsány skriptovací jazyk PHP, JavaScript s knihovnou jQuery a databázový systém MySQL. Téma je rozsáhlé a daleko převyšuje možnosti této práce, proto jsou vybrána a popsána jen některá podtémata, která jsou ve výsledné aplikaci použita.

### 2.4.1 JavaScript a jQuery

JavaScript je objektově orientovaný skriptovací jazyk, který je nezávislý na platformě a syntakticky spadá do rodiny jazyků C, C++ a Java. Méně známé pojmenování jazyka je ECMAScript a je vyvíjen mezinárodní asociací Ecma. Je to dynamicky typovaný a funkcionální jazyk (funkce lze do sebe vnořovat). Jazyk je často využíván pro tvorbu dynamických webových stránek, pak hovoříme o DHTML (Dynamic HTML). Skripty jsou vykonávány na straně klienta a používány k ovládání interaktivních prvků, jako jsou vstupy ve formulářích, tlačítka a podobně. Pomocí JavaScriptu lze také vytvářet animace, manipulovat s grafickým rozložením stránek pomocí CSS (Cascading Style Sheet) atd. Oficiální dokumentace je dostupná on-line na stránkách asociace Ecma (2014). Kód JavaScriptu může být umístěn přímo v dokumentu HTML (HyperText Markup Language) nebo v externím

souboru, který je k HTML dokumentu připojen. Implementace některých skriptů je pro různé prohlížeče rozdílná a vývojář musí psát více kódu. Řešením tohoto problému jsou knihovny, které vnitřně řeší rozdílnost implementace, o kterou se vývojář nemusí nadále starat. Často využívanou JavaScript knihovnou je jQuery. Tuto knihovnu při vývoji použil autor této práce a je dále popsána.

Používání knihoven jako je jQuery přináší z pohledu vývojáře několik výhod. Jak již bylo zmíněno výše, jQuery sjednocuje implementaci některých algoritmů, dále zjednodušuje zdrojový kód, čímž omezuje prostor pro chyby, kterých se vývojář může dopustit. Knihovna a dokumentace je dostupná on-line na stránkách asociace jQuery (JQUERY, 2014). Knihovna obsahuje funkce pro práci s událostmi, DOM (Document Object Model), AJAX (Asynchronous JavaScript and XML) a animace. Umístění knihovny je možné dvojím způsobem. Knihovnu lze celou stáhnout a umístit na web server souběžně s dokumentem HTML, nebo lze v hlavičce dokumentu HTML odkázat na knihovnu umístěnou na serveru společnosti Google nebo jiného poskytovatele této knihovny. Funkce jQuery jsou volány klíčovým slovem `jQuery` nebo zkráceně znakem `$`. Knihovna jQuery využívá selektory odvozené z CSS, čímž usnadňuje orientaci v kódu. Například následující kód vybere všechny elementy s třídou `HelloWorld` a při kliknutí spustí anonymní funkci, která elementy naplní HTML kódem, který tučně zobrazí text `Hello World`.

```
$(`.HelloWorld`).click(function(){  
    $(`.HelloWorld`).html("<b>Hello World</b>");  
})
```

## 2.4.2 PHP

Hypertext Preprocessor, nebo PHP, je skriptovací jazyk, často používaný ke generování dynamického webu ale i pro desktopové aplikace. Pro webové použití jsou skripty vykonávány na straně serveru a klientovy je odeslán výsledek. Popis a dokumentace jazyka PHP je dostupná on-line z domovských stránek <http://www.php.net/>.

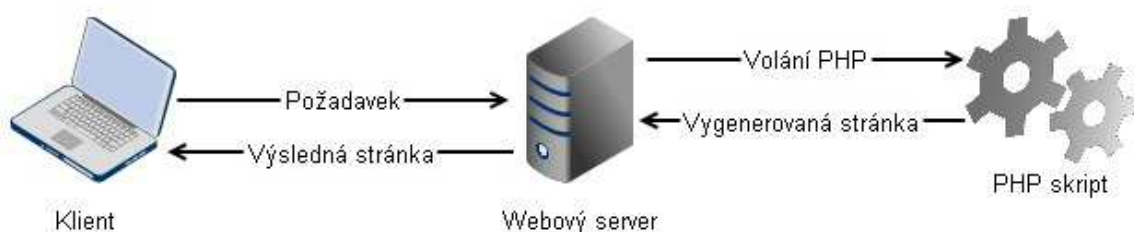
K propojení PHP skriptů a HTML stránky je možné použít několik přístupů. Jedním přístupem je zapisování PHP kódu přímo do zdrojového kódu stránky. V následujícím příkladu se PHP stará o zobrazení jména přihlášeného uživatele, které je uloženo v globální proměnné `$_SESSION`. V tomto případě se kód PHP skriptu prolíná se značkovacím kódem HTML.

```
<div class='userControl'>  
    <?php if (isset($_SESSION['user'])){ ?>  
        Přihlášen uživatel: <?php echo ($_SESSION['user']) ?>  
    <?php }else{ ?>  
        Přihlašte se  
    <?php } ?>  
</div>
```



Na serveru se provede PHP kód, který rozhodne, jaká větev podmínky se vykoná. Je-li v proměnné `$_SESSION` uloženo jméno uživatele výsledný dokument vypíše jeho jméno. V opačném případě vyzve uživatele aby se přihlásil. Posloupnost úkonů při generování stránky je ilustrována na obrázku 1. Skript vygeneruje stránku, předá ji web serveru a ten ji celou odešle klientovi.

**Obr. č. 1 Generování webové stránky** (zdroj: vlastní tvorba)



Výsledný kód stránky na straně klienta nebude obsahovat žádný PHP kód.

Dalším přístupem je používání technologie AJAX. Tento přístup je složitější ale poskytuje uživateli dojem jako z desktop aplikace a v určitých situacích snižuje celkový přenesený objem dat. AJAX není samostatnou technologií ale novým způsobem jak jsou stávající technologie využity. Ke komunikaci mezi klientem a serverem dochází na pozadí stránky a při obdržení dat ze serveru není nutné překreslovat celou stránku. O doplnění dat do stránky se stará skript na straně klienta, který odesílá a přijímá data ze serveru, tudíž není třeba generovat celou aktualizovanou stránku na serveru. Tento přístup výměny dat umožňuje oddělit PHP skripty od HTML stránky a zdrojový kód je přehlednější a působí čistším dojmem. Aplikace často kombinují výhod obou přístupů.

Skripty PHP se také často starají o komunikaci s databází. Dnešním trendem je používat takzvané parametrizované funkce, které zvyšují přehlednost a bezpečnost aplikace. Dotazy a samotná data jsou databázi předány odděleně a o jejich dosazení se stará sama databáze. Tuto konstrukci ilustruje následující část kódu.

```
$insertUser = $pdo->prepare(
    "INSERT INTO 'users' ( 'name', 'password' ) VALUES ( ?, ? )";
$insertUser->execute(array('JmenoUzivatele', 'qwert'));
```

Na místo proměnných jsou v dotazu použity otazníky a proměnné jsou předány později v jednom poli. Parametrizované funkce třídy `PDOStatement` jsou detailně popsány v oficiální dokumentaci jazyka PHP (2014).

### 2.4.3 MySQL

MySQL je relační databázový systém, s nímž probíhá komunikace pomocí jazyka SQL (Structured Query Language) a je volně šiřitelný. Dokumentace, návody a samotný databázový systém je

dostupný na oficiálních stránkách výrobce ORACLE (2014). Data jsou uložena v tabulkách tvořených řádky a sloupce. Řádky reprezentují jednotlivé záznamy a sloupce atributy záznamů. Vazby mezi tabulkami jsou tvořené relacemi a realizované cizími klíči. Cizí klíč je sloupec nebo skupina sloupců v jedné tabulce, které definují vazbu na další tabulku. Podle návrhu tabulek případně struktury dat se rozlišují typy vazeb. Vazba 1 : 1 nastává ve chvíli, kdy jeden záznam v jedné tabulce odpovídá právě jednomu záznamu v jiné tabulce. Nicméně tato vazba se obvykle nevyskytuje, z hlediska uložení dat nemá moc smysl a je principiálně správné tabulky sloučit. Častější situací je vazba 1 : n, záznamu v jedné tabulce odpovídá několik záznamů v jiné tabulce. Další vazbou je m : n, několika záznamům v jedné tabulce lze přiřadit několik záznamů v jiné tabulce.

Databázový systém MySQL disponuje rozšířením pro prostorová data Spatial Extensions a podporuje prostorové datové typy. Výměnnými formáty jsou WKT (Well Known Text) nebo WKB (Well Known Binary), které původně definovalo Open Geospatial Consortium (HERRING, 2011). Databázový systém disponuje sadou funkcí pro manipulaci s prostorovými daty. Jde například o výpočet obsahu prvků, překrývání, slučování, průnik, buffer prvků a mnoho dalšího (ORACLE, 2014).

SQL dotazy se dělí do několika skupin podle toho, jak nakládají s daty. V aplikaci jsou použity příkazy pro manipulaci s daty např. INSERT, SELECT, DELETE atd. a pro definici dat CREATE, ALTER, DROP atd. Jazyk byl navržen tak, aby příkazy co nejvíce připomínaly lidskou řeč a byly srozumitelné. Například následující kód vybere z tabulky users jména a příjmení uživatelů, kteří se narodili v Olomouci a nejsou v produktivním věku.

```
SELECT name, surname FROM users WHERE birthplace = Olomouc AND age NOT  
BETWEEN 15 AND 64
```

Dotazy a tabulky se dají různě kombinovat a vzniká tak mocný nástroj pro těžení informací z dat

## 3 METODIKA A REALIZACE APLIKACE

Jak již bylo zmíněno v úvodu kapitoly 2, hlavním cílem této práce je návrh a realizace webového nástroje pro sběr, evidenci a validaci geografických dat. V kapitole 3 bude rozebrán postup při vývoji aplikace od návrhu po samotnou realizaci. Následující text může sloužit jako stručný a obecný návod k vytvoření webové aplikace.

### 3.1 Návrh aplikace

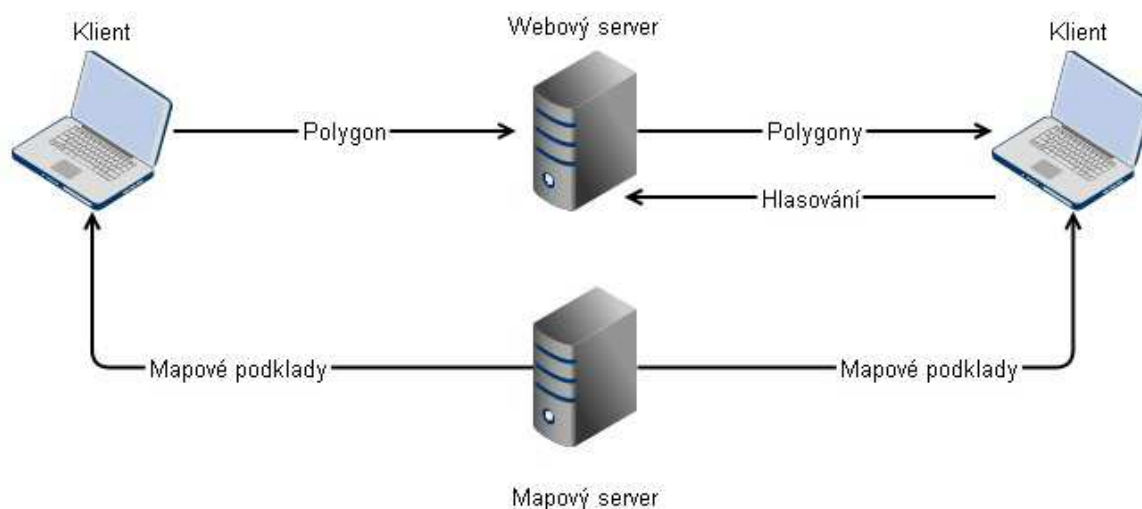
Klíčem k úspěchu při vývoji jakékoli aplikace je její celkový návrh. Před samotnou realizací by měla být navržena vnitřní logika aplikace, grafické komponenty, uživatelské rozhraní, a datové úložiště. V následujících oddílech je popsán návrh aplikace, jak byl autorem prezentován před započatím vývoje.

#### 3.1.1 Logika aplikace

Obrázek č. 2 ilustruje základní funkce aplikace, které jsou:

- Zobrazování on-line mapových podkladů.
- Vkládání polygonů, které reprezentují chyby v land-use mapě.
- Hlasování o správnosti (polohové, tematické) vložených polygonů.

Tyto tři funkce předpokládají celou řadu dílčích funkcí, bez kterých se chod aplikace neobejde, a vyžadují komplexní řešení. Dále jsou uvedeny některé dílčí funkce, jejichž počet se při samotném vývoji bude přirozeně rozrůstat s tím, jak aplikace bude nabývat na komplexnosti. Všichni uživatelé musejí být registrovaní a disponovat jednoznačným identifikátorem. Tato podmínka umožňuje omezit uživatele tak, aby mohl pro každý polygon hlasovat jen jednou, a umožňuje zaznamenat jméno autora při vložení nového polygonu. Je nutné omezit výběr kategorií, ze kterých uživatel vybírá při tvorbě nového polygonu. Tímto se předejde vkládání libovolného popisu kategorie a je zajištěna celková harmonie vložených dat pro případné následující zpracování.

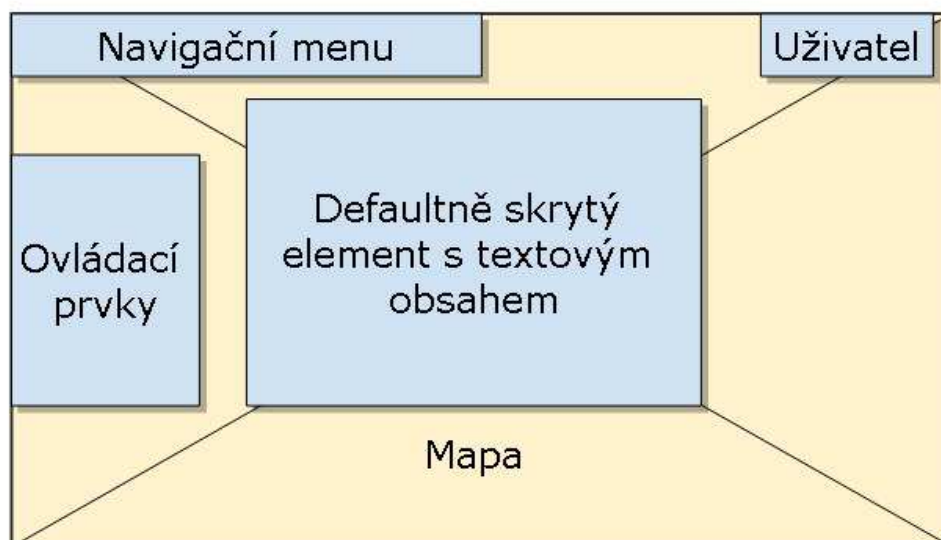
**Obr. č. 2 Diagram základní funkčnosti aplikace** (zdroj: vlastní tvorba)

Uvedená funkcionality vyžaduje sadu technologických prostředků, které jsou popsány výše. Klientské skripty budou psány v jazyce JavaScript za pomoci knihovny jQuery, serverové skripty v jazyce PHP a data budou uložena v databázovém systému MySQL.

### 3.1.2 Grafický výstup a uživatelské rozhraní

Grafický návrh se může zdát nadbytečný, týká-li se odborných prezentací nebo nástrojů. Ani cílem této práce není vytvořit produkt na úrovni profesionálního webového designu, přesto si toto téma zaslouží několik slov i zde, protože grafika je nedílnou součástí každé webové, potažmo vizuální prezentace. Grafika webové aplikace je navázána na uživatelské rozhraní a to je dále navázáno na vnitřní logiku aplikace, proto alespoň základní grafický návrh pomáhá vývojáři při práci a poskytuje mu obecnou představu jak propojit logickou stránku aplikace s grafickou.

Koncepce webových stránek je navržena s jednou základní stránkou, kde bude umístěn veškerý obsah a samotná mapová aplikace. Další podstránky budou pouze pro vedlejší funkce, jako je aktivace uživatelských účtů. Na obr. 3 je zobrazeno základní schéma hlavní stránky. V levé horní části je umístěno navigační menu, které uživateli umožňuje zobrazit elementy s textovým obsahem a návrat k mapě. Po levé straně je panel, který obsahuje ovládací prvky aplikace. V pravém horním rohu je umístěn panel, určený pro práci s uživatelským účtem (přihlašování, registrace, odhlašování).

**Obr. č. 3 Grafický návrh** (zdroj: vlastní tvorba)

Detailní rozložení ovládacích prvků a podrobný grafický výstup není třeba v návrhu uvádět, protože v průběhu vývoje aplikace bude pravděpodobně docházet k drobným úpravám.

### 3.1.3 Datové úložiště

K ukládání dat se nabízejí dva základní přístupy. Data je možné ukládat přímo do souboru, který je umístěn na web serveru nebo do databáze na databázovém serveru. S postupným užíváním aplikace bude narůstat datový objem, bude docházet k manipulaci s daty a jejich vytěžování. Z principu je ukládání do souboru nevhodné a bude použita databáze.

Pro uložení dat bude využit databázový systém MySQL a pro běh aplikace bude třeba tří tabulek. První tabulka bude obsahovat veškeré informace o uživateli. Následuje výčet atributů, v uvozovkách je název atributu:

- Unikátní identifikátor „ID“ – datový typ int
- Uživatelské jméno „nick“ – datový typ varchar
- Uživatelské heslo „password“ – datový typ char
- Tzv. sůl „salt“ – datový typ varchar
- E-mail uživatele „mail“ – datový typ varchar
- Oprávnění uživatele „status“ – datový typ tinyint
- Čas a datum registrace „timestamp“ – datový typ timestamp

Uložení polygonů vyžaduje dvě tabulky. První tabulka bude obsahovat kategorie a jejich identifikátor. Takto oddělený popis kategorií usnadňuje následnou editaci a práci s daty.

- Unikátní identifikátor „ID“ – datový typ int
- Název kategorie „category“ – datový typ varchar

Druhá tabulka týkající se polygonů bude obsahovat veškeré informace o vložených polygonech. Cizími klíči bude propojena s tabulkami kategorií a uživatelů.

- Unikátní identifikátor „ID“ – datový typ int
- Čas a datum vložení polygonu „timestamp“ – datový typ timestamp
- Identifikátor kategorie „categoryId“ – datový typ tinyint
- Součet hlasování „vote“ – datový typ int
- Polohové určení polygonu „vertices“ – datový typ polygon
- Identifikátor uživatelů, kteří již hlasovali „voters“ – datový typ varchar
- Identifikátor uživatele, který polygon vytvořil „user“ – datový typ int

### 3.1.4 Datové zdroje

Aplikace bude vyvíjena na základě Google Maps API. Díky této API budou v aplikaci využity základní topografická a satelitní mapa Google. Mapy Google používají Merkátorovo úhlojevné válcové zobrazení a zeměpisný souřadnicový systém WGS 84. Dalšími vstupními daty je katastrální mapa, připojená pomocí WMS. Katastrální mapa je původně v Křovákově zobrazení v systému JTSK. Při připojení webové mapové služby katastrální mapy lze vybrat v jakém zobrazení bude vracena z mapového serveru. V aplikaci je zvolen stejný souřadnicový systém jako má základní mapa, WGS 84. Katastrální mapa je rozdělena do jednotlivých dlaždic, které jsou klientovy poskytovány postupně. Tento přístup je výhodný z pohledu přenosu dat, protože z mapového serveru jsou přenášeny vždy jen ty části mapy, které jsou v aktuálním náhledu.

Posledním zdrojem dat, s kterým aplikace bude pracovat, jsou uživateli vložené polygony, které jsou následně distribuovány ze serveru v souborech formátu KML, což je aplikace formátu XML a je standardem Open Geospatial Consortium (WILSON, 2008). Soubory KML budou vytvořeny skriptem na serveru a odeslány klientovi, kde budou dále parsovány objektem třídy KmlLayer. Celkový datový tok a komunikace se servery je vyjádřena v následujících, po sobě jdoucích krocích:

- Klient odešle požadavek na web server
- Web server obdrží požadavek a odešle zpět vygenerovanou stránku
- Klient obdrží stránku
- Je staženo API do klienta
- Klient odešle požadavek mapovému serveru Google
- Je inicializována mapa a poskytnuta klientovi
- Je načten soubor KML, parserem vytvořeny polygony
- Jsou zobrazeny polygony v mapě

Polygonům, které uživatelé vkládají, bude možné přiřadit jednu z deseti kategorií, které vycházejí z katastrální mapy. Tento seznam lze kdykoliv rozšířit, například o kategorie nově implementovaného zdroje land use dat. Následuje výčet druhů pozemku, které aplikace využije:

- orná půda
- chmelnice
- vinice
- zahrada
- ovocný sad
- trvalý travní pozemek
- lesní pozemek
- vodní plocha
- zastavěná plocha
- ostatní plocha

### 3.1.5 Bezpečnost aplikace

Bezpečnost je rozsáhlé téma, které je od vývoje aplikací neoddělitelné. Zejména on-line aplikace jsou z bezpečnostního hlediska citlivé, protože k nim mají přístup všichni uživatelé internetu. Někteří uživatelé mohou ale přistupovat k aplikaci s úmyslem škodit, jedná se o tzv. hackery nebo škodlivé aplikace. Mezi časté typy útoků se řadí SQL injection a cross site scripting.

Principem útoku SQL injection je využívání nezabezpečené databáze. Útočník do vstupního formuláře aplikace vloží vlastní SQL dotaz, který je následně vykonán na serveru. Útočník takto může získat, manipulovat nebo mazat data z databáze. Chránit aplikaci proti tomuto typu útoku lze například používáním parametrizovaných dotazů, které byly popsány v kapitole 2.4.2. Dotaz a data jsou od sebe oddělena a útočníkem vložený dotaz je vždy použit jako datový vstup nikoliv dotaz. Veškeré SQL dotazy ve výsledné aplikaci budou konstruovány pomocí parametrizovaných dotazů, takže data jsou proti útoku SQL injection chráněna.

Útok XSS (cross site scripting) je metoda útoku, kdy útočník vkládá své skripty nebo odkaz na ně do nezabezpečených stránek. Proti tomuto útoku se dá bránit filtrováním vložených dat. V aplikaci bude pro všechny vstupy použita funkce `filter_input`, která z dat odstraní potenciálně škodlivý obsah.

Přes veškerá opatření stále existuje riziko, že útočník získá data z databáze. V takovém případě jsou nejcitlivější data uživatelská hesla, proto jsou ukládány jejich otisky, ze kterých nelze v rozumném čase zjistit původní heslo. Tato metoda je známa jako hashování a je často kombinována s přidáváním tzv. „soli“. „Sůl“ je náhodný řetězec znaků, který je přidán k heslu před samotným hashováním. Takto je řešena situace, kdy dva uživatelé mají stejné heslo, ale výsledný otisk nikoliv. Po úspěšné registraci je do databáze uložen pouze otisk „osoleného“ hesla a samotná „sůl“. Když se

uživatel přihlašuje, je nejprve získána příslušná „sůl“ z databáze, vytvořen otisk vloženého hesla a následně porovnán s otiskem uloženým při registraci. Výsledek je takový, že skutečné heslo není nikde uloženo a nemůže být odcizeno. V naší aplikaci bude použit pro hashování hesel algoritmus sha512, který je v současné době považován za bezpečný.

## 3.2 Zdrojový kód aplikace

Hlavní stránkou, ve které se setkávají a vzájemně spolupracují téměř všechny skripty aplikace, je soubor index.php. Tento soubor je sestaven z HTML tagů, které dávají výsledné stránce strukturu, PHP skriptů, které poskytují různý výstup pro přihlášené a anonymní uživatele, a v elementu head jsou připojeny soubory s klientskými skripty. V následujících oddílech budou podrobně rozebrány a popsány skripty, ze kterých je aplikace sestavena. Pro snazší pochopení následujících odstavců doporučuje autor nahlížet do kompletního zdrojového kódu. Některé části skriptů nejsou níže popsány nebo jsou zkráceny. Jedná se především o konfigurační soubor pro připojení k databázi, který z bezpečnostních důvodů není možné zveřejnit. Dále není popsáno připojení k databázi, které je pro všechny skripty stejné a pro pochopení aplikační logiky nevyžaduje vysvětlení. Také nejsou popsány některé funkce, které mají na starost grafiku aplikace, která je pro tuto práci druhořadá.

### 3.2.1 Mapové podklady

Zobrazování a práce s mapou je zajištěna pomocí Google Maps API. Nejprve je v hlavičce dokumentu index.php vložena adresa pro stažení API:

```
<script  
src="http://maps.googleapis.com/maps/api/js?key=AIzaSyCryEeFiSuglxeZ-  
fW3Jp2L_JnXEpk2auY&sensor=false&libraries=drawing"></script>
```

V URL adrese je několik přidáných proměnných. Proměnná key obsahuje jedinečný klíč, který byl vygenerován při zakládání nového projektu. Proměnná sensor definuje schopnost aplikace využívat sensor pro detekci polohy (GPS). Proměnná libraries definuje, jaké přídatné knihovny aplikace používá a vyžaduje při stahování API.

Další skripty týkající se mapových podkladů jsou v odděleném souboru mashUpMap.js. Část skriptu, který se provádí ihned po načtení stránky je ve funkci initialize(). Jako první je proměnná mapProp, která nastavuje základní mapu a její ovládací prvky. Dále je vytvořen objekt map, jehož první parametr určí, v jakém elementu bude mapa vykreslena, a v druhém parametru je nastavení mapy mapProp.

```
var mapProp = {  
  center: new google.maps.LatLng(mapLat,mapLng), zoom: mapZoom,  
  mapTypeId: google.maps.MapTypeId.ROADMAP,  
  mapTypeControlOptions: {
```



```

        style: google.maps.MapTypeControlStyle.HORIZONTAL_BAR,
        position: google.maps.ControlPosition.BOTTOM_LEFT},
    zoomControlOptions: {
        style: google.maps.ZoomControlStyle.LARGE,
        position: google.maps.ControlPosition.LEFT_BOTTOM
    }
  });
map = new google.maps.Map(document.getElementById("mapField"), mapProp);

```

Z důvodu asynchronního načítání mapy, například při odeslání nového polygonu, jsou proměnné `mapLat`, `mapLng` a `mapZoom`, které se starají o pozici a přiblížení mapy, deklarované mimo funkci `initialize()`. Takto mohou k proměnným přistupovat i jiné funkce.

Další mapovou vrstvou je katastrální mapa připojená pomocí WMS. Google Maps API nepodporuje WMS přímo a tyto vrstvy je třeba připojit ručně, jako překryvnou vrstvu obrazů. V objektu `CUZKwms`, třídy `ImageMapType`, je sestavena URL adresa každé požadované dlaždice a následně je objekt přidán do mapy metodou `push`. URL dlaždice definuje požadavek na mapový server a obsahuje veškeré údaje potřebné pro zobrazení vrstvy. Za zmínku stojí parametr `LAYERS`, ve kterém jsou požadované vrstvy katastrální mapy (např. parcelní čísla, hranice parcel).

```

var CUZKwms = new google.maps.ImageMapType({
  getTileUrl: function(coord, zoom) {
    var url = "http://services.cuzk.cz/wms/wms.asp?";
    url += "&REQUEST=GetMap";
    url += "&SERVICE=WMS";
    url += "&VERSION=1.1.1";
    url += "&LAYERS=parcelni_cisla_i,hranice_parcel_i,obrazy_parcel_i";
    url += "&STYLES=";
    url += "&SRS=EPSG:4326";
    return url;
  },
  tileSize: new google.maps.Size(256, 256),
  isPng: true});
map.overlayMapTypes.push(CUZKwms);

```

Implementace WMS vrstvy v prostředí Google Maps API byla převzata a modifikována ze zdrojového kódu projektu Katastr nemovitostí 2.0 (URL: <http://www.katastr2.cz/>).

Poslední vrstva, která se zobrazuje v mapě, jsou polygony, které vložili uživatelé. Polygony přidává do mapy objekt `kmlFromDB` třídy `KmlLayer`. Tomuto objektu jsou předávány soubory KML, které generuje samostatný skript na serveru. Na tento skript odkazuje URL v proměnné `kmlSource`. Protože mapové servery společnosti Google ukládají KML soubory do své cache paměti a při opakovaném volání stejného souboru použijí soubory ze své paměti, nikoliv ze zadané URL adresy, nejsou výsledné polygony aktualizovány. Tomuto lze předcházet tím, že k URL adrese je přidán

parametr s aktuálním časem a mapový server musí soubor s polygony aktualizovat, protože se jeho jméno neshoduje se jménem souboru v cache paměti.

```
var kmlSource = 'http://bp.test.cernik.name/genXML.php?t=' + (new
Date()).getTime();
kmlFromDB = new google.maps.KmlLayer({
    url: kmlSource,
    map: map});
```

### 3.2.2 Uživatelské účty

Pro webovou aplikaci tohoto typu je nezbytné registrovat uživatele. Tato operace vyžaduje komplexní sadu skriptů, jak na straně klienta, tak na straně serveru. Následující odstavce popisují skripty a logickou strukturu v pořadí, ve kterém je manipulováno s daty uživatelských účtů. Nejprve bude popsána registrace uživatelů na portálu, následně přihlašování a odhlašování.

Při registraci je uživatel vyzván, aby vyplnil své uživatelské jméno, e-mail a heslo. Před samotným odesláním těchto údajů na server dochází na straně klienta k několika úkonům, o které se starají skripty v souboru user.js. Vkládá-li uživatel svoje jméno, dochází při každé události keyup (když uživatel napíše písmeno) k odeslání vloženého textu na server. Data jsou odesílána a přijímána asynchronně metodou post z knihovny jQuery. Pokud server vrací hodnotu true je zobrazen pod registračním formulářem element s upozorněním, že vložené jméno je již obsazené. Pokud je vrácena hodnota false je upozornění skryto. Souběžně je upravena hodnota proměnné regFormValid, která je využita dál ve skriptu.

```
$("#nick").keyup(function() {
    $("#emptyNick").hide();
    var nick = $("#nick").val();
    $.post("vali.php", {nick: nick}, function(data) {
        if (data) {$("#wrongNick").show();
            regFormValid = 0;
        } else {$("#wrongNick").hide();
            regFormValid = 1;}
```

Skript na serveru v souboru vali.php přebírá data, navazuje spojení s databází a odesílá do databáze dotaz na uživatele, kteří mají takové uživatelské jméno, jaké vložil registrující se uživatel. Když databáze takový záznam najde, uloží se do proměnné \$result výsledek hledání, v opačném případě zůstává proměnná prázdná. Podle výsledku je příkazem echo odeslána zpět klientovi hodnota true nebo false.

Dále jsou v souboru user.js dvě anonymní funkce, které skrývají upozornění o špatně vyplněném e-mailu a o nevyplněném heslu a nastavují proměnnou regFormValid na hodnotu 1.

Následuje funkce, která je spuštěna když uživatel potvrdí registrační formulář. Nejprve proběhne série kontrol validnosti zadaných hodnot. Pokud je nějaká hodnota v nepořádku, zobrazí se příslušné upozornění a proměnná `regFormValid` je nastavena na hodnotu nula.

```
$("#potvrd").click(function() {
    if (!nick) {$("#emptyNick").show();
        regFormValid = 0;}
    if (!checkMail(mail) && mail) {wrongMail().show();
        regFormValid = 0;}
    if (!mail) {$("#emptyMail").show();
        regFormValid = 0;}
    if (!pass1 || !pass2) {$("#emptyPass").show();
        regFormValid = 0;
        $("#pass1, #pass2").val("");}
```

Dále se skript větví podle toho, na jakou hodnotu je nastavena proměnná `regFormValid`. Pokud výsledek předešlých kontrol je v pořádku (`regFormValid == 1`), dojde k asynchronnímu odeslání registračních dat na server, v opačném případě je uživatel upozorněn na chyby ve formuláři probliknutím zobrazených upozornění. Podle vrácených dat ze serveru je skript rozdělen do čtyř větví.

- Uživatel je informován o úspěšné registraci, formulář je vymazán a skryt.
- Uživatel je informován o obsazeném jménu, jméno a heslo jsou z formuláře vymazána.
- Uživatel je informován o neshodě kontrolního hesla, hesla jsou smazána.
- Uživatel je informován o selhání registrace z neznámého důvodu, formulář je vymazán.

```
if (regFormValid) {
    $.post("userToDB.php", {nick: nick, mail: mail, pass1: pass1, pass2:
pass2}, function(data) {
        // zpracování vrácených dat - viz kompletní zdrojový kód
    }else{$("#invalidReg:visible").delay(100).fadeOut().fadeIn('fast');}
```

Na serveru obdrží a zpracuje data skript v souboru `userToDB.php`. V případě, že by kontrola obsazenosti jména v klientském skriptu z jakéhokoliv důvodu selhala, je zde kontrola provedena znovu. Pokud je výsledek kontroly v pořádku a heslo souhlasí s kontrolním heslem, je vykonán následující skript. Je vygenerován náhodný řetězec (tzv. „sůl“), přidán k heslu a z výsledku je vytvořen otisk algoritmem `sha512`. Tento postup je výše rozebrán v podkapitole 3.1.5 o bezpečnosti aplikace. Dále je odeslán příkaz databázi, která uloží údaje o uživateli do tabulky `users`, včetně otisku hesla. Když proběhne uložení dat do databáze, je vytvořen aktivační e-mail s přihlašovacími údaji a odeslán na e-mailovou adresu uživatele.

```
if($pass1 == $pass2){
    $salt = uniqid();
    $hash = hash('sha512', $pass1 . $salt);
```

```

$insertUser = $pdo->prepare("insert into users (nick, password,
salt, mail) values (?, ?, ?, ?)");
$insertUser->execute(array($nick, $hash, $salt, $mail));
$insertUserResult = $insertUser->rowCount();
if ($insertUserResult == 1){
    // generování e-mailu - viz kompletní zdrojový kód
    if(mail($to, $subject, $message, $headers)){ echo 0;
    }else{echo 1; // mail posting failed}
}else{echo 1; //registration failed}
}else{echo 2; // passwords do not match}

```

Takto registrovaný uživatel se může přihlásit a prohlížet informace o polygonech, ale nemá dostatečné právo na vkládání nových polygonů a hlasování. Pro rozšíření práv musí uživatel otevřít odkaz, který mu byl zaslán v e-mailu. Tato URL adresa směřuje na aktivační stránku a v parametrech nese otisk uživatelského jména a ID. Skript, který generuje aktivační stránku získá hodnoty z parametrů v URL a porovná je s údaji v databázi. Pokud si údaje odpovídají, změní v databázi status uživatele na hodnotu 1, čímž uživatel získá plná práva. Následně je vygenerována stránka s obsahem podle toho, jestli došlo k aktivaci nebo aktivace selhala.

Přihlašování uživatelů probíhá ve formuláři na hlavní straně. Vyplněné údaje, jméno a heslo, jsou odeslány na server metodou POST, kde je zpracuje skript v souboru login.php. Nejprve je z databáze získána „sůl“ přihlašujícího se uživatele. Tato „sůl“ je spojena s vloženým heslem a stejně jako při registraci je výsledný text zahashován algoritmem sha512. Následně je databázi odeslán dotaz, zda existuje položka s odpovídajícím jménem a heslem. Pokud je odpověď kladná, dojde k nastavení proměnné \$\_SESSION. Jedná se o globální proměnnou, určenou k přenášení dat mezi jednotlivými stránkami. Je do ní uloženo jméno, ID a status uživatele. Následně dojde k přesměrování na hlavní stránku, k URL adrese je přidán parametr login, a nastaven na hodnotu 1. Pokud není uživatel nalezen, dojde ke zničení session a parametr URL je nastaven na 0.

```

$salt = $pdo->prepare("select salt from users where nick =?");
$salt->execute(array($userName));
$userSalt = $salt->fetch();
$hash = hash('sha512', $userPass . $userSalt[0]);
$query = $pdo->prepare("select * from users where nick=? and password=?");
$query->execute(array($userName, $hash));
$resultUser = $query->fetch();
if($resultUser){
    // nastavení session a přesměrování - viz kompletní zdrojový kód

```

Po přesměrování na hlavní stránku index.php je vyhodnocen parametr login a vytvořena výsledná stránka podle toho, zda došlo k přihlášení nebo ne. Pokud dojde k přihlášení, jsou před samotnou deklarací doctype a strukturou stránky do lokálních proměnných \$user, \$userID a

\$userStatus uloženy hodnoty z globální proměnné \$\_SESSION. Pokud k přihlášení nedošlo je session zničena. V pravém horním panelu je po přihlášení zobrazeno jméno přihlášeného uživatele a tlačítko k odhlášení. Pokud není uživatel přihlášen, jsou v tomto panelu tlačítka pro vyvolání přihlašovacího a registračního formuláře.

K odhlášení dojde po stisknutí odhlašovacího tlačítka v pravém horním rohu, které směřuje na soubor logout.php. Tento skript vyprázdní proměnnou \$\_SESSION, zničí session a přesměruje na hlavní stranu.

### 3.2.3 Vkládání polygonů

Aplikace má dva základní módy pro přihlášené uživatele. K přepínání mezi těmito módy jsou určena tlačítka v záhlaví levého výsuvného panelu. Aplikace je defaultně v módu vybírání polygonů. O přepnutí se stará skript v souboru mashUpMap.js, který způsobí, že po stisknutí tlačítka s ID menuBtnsSelect lze zakreslovat polygony do mapy a zároveň je zobrazen panel se select formulářem, kde lze vybrat kategorii z katastrální mapy.

```
var menuBtnsSelect = document.getElementById("menuBtnsSelect");
google.maps.event.addDomListener(menuBtnsSelect, 'click', function() {
    selectMode();
    drawingManager.setDrawingMode(google.maps.drawing.OverlayType.NULL);
```

Funkce selectMode() je umístěna v souboru layout.js a ovládá některé grafické funkce, které jsou vykonány po přepnutí módu. V této funkci je volána funkce delPlg(), která je umístěna v souboru mashUpMap.js a smaže již nakreslené nebo rozkreslené polygony. Funkce znemožňuje v tomto případě zakreslit dva polygony najednou.

```
function delPlg() {
    drawingManager.setDrawingMode(null);
    plg.setMap(null);
    plg.length = 0;
    drawingManager.setDrawingMode(google.maps.drawing.OverlayType.POLYGON);
```

Zakreslování polygonů má na starost objekt drawingManager. Při jeho inicializaci je nastavena viditelnost jeho výchozího ovládání na false a možnost editace zakreslených polygonů na true.

```
drawingManager = new google.maps.drawing.DrawingManager({
    drawingMode: google.maps.drawing.OverlayType.NULL,
    drawingControl: true,
    map: map,
    polygonOptions: {editable: true},
    drawingControlOptions: false});
```

Objektu `DrawingManager` je přiřazen naslouchač události ukončení kreslení polygonu. Po této události je nakreslený polygon uložen do proměnné `plg` a mód kreslení je vypnut. Jde tedy o událost kdy uživatel uzavře kreslený polygon.

```
google.maps.event.addListener(drawingManager, 'overlaycomplete',  
function(event) {  
    plg = event.overlay;  
    plg.type = event.type;  
    drawingManager.setDrawingMode(null);});
```

Nyní uživatel zadá v select formuláři kategorii vkládaného polygonu a tlačítkem odešle polygon. V tuto chvíli je spuštěna funkce `uploadPlg()` v souboru `mashUpMap.js`. Tato funkce nejprve zkontroluje status uživatele, zda má dostatečné oprávnění pro vkládání polygonu. Následně je ve for cyklu uložen polygon ve formátu WKT do proměnné `polygonWKT`. Je získána kategorie polygonu z formuláře a ID uživatele, které bylo uloženo do proměnné `userID` při přihlašování. Poté jsou data asynchronně odeslána na server metodou `post` a podle odpovědi ze serveru se skript větví. Pokud je odpověď kladná a polygon byl uložen, je uživatel informován, aplikace přepnuta do select módu výše popsanou funkcí `selectMode()` a asynchronně obnovena mapa funkcí `refreshMap()`. Pokud by nebyla zadaná kategorie nebo polygon, je uživatel pouze informován a k odeslání dat nedojde.

```
function uploadPlg() {  
    if (checkStatus()) {var category = $("#category").val();  
        if (category != 0 && plg.length != 0){  
            var polygonWKT = 'POLYGON(';  
            //uložení polygonu v cyklu - viz kompletní zdrojový kód  
            $.post('plgToDb.php', {lngLat: polygonWKT, category: category,  
userID: userID}, function(data) {  
                if (data) {myAlert('Polygon byl uložen.');                    refreshMap();  
                    selectMode();  
                } else {myAlert('Uložení polygonu selhalo.')}});  
        } else {myAlert('Zadejte polygon a kategorii.');    }  
}
```

Funkce `refreshMap()` nejprve uloží do proměnných aktuální polohu a přiblížení mapy, aby došlo k obnově v současném náhledu mapy.

```
function refreshMap(){  
    mapZoom = map.getZoom();  
    mapLat = map.getCenter().lat();  
    mapLng = map.getCenter().lng();  
    asynLoadingMap();}
```

Na konci funkce `refreshMap()` je zavolána funkce `asynLoadingMap()`, která asynchronně načte mapy a nemusí být překreslena celá stránka. Dojde k vytvoření skriptu, který obsahuje URL pro

načtení API a mapy, a jako callback funkce (funkce, která je provedena po dokončení skriptu) je volána celá funkce `initialize()`, které jsou ale předány nové proměnné polohy a přiblížení.

Data s polygonem zpracuje na serveru skript v souboru `plgToDb.php`. Tento skript jednoduše odešle příkaz a data databázi. Do tabulky `polygons` je uloženo ID uživatele který polygon vytvořil, kategorie polygonu a samotný polygon ve formě WKT. Takto zapsaný polygon převede databáze sama do datového formátu `polygon`.

### 3.2.4 Vykreslování polygonů a stahování KML

V oddíle 3.3.1 Mapové podklady je popsáno jak jsou polygony renderovány v mapě z KML souborů. Níže v tomto oddíle je popsáno, jak vzniká zmiňovaný KML soubor a jak probíhá stahování souborů uživatelem.

Vždy, když je mapa obnovena stahuje objekt `kmlFromDB` soubor s polygony ze serveru. Tento soubor je vždy vygenerován znovu a tím je zajištěna aktuálnost zobrazovaných polygonů. Soubor s polygony je generován skriptem v souboru `genXML.php`. Nejprve jsou z databáze získány všechny polygony. Pomocí cizích klíčů jsou v SQL dotazu připojeny tabulky `users` a `category`, takže výsledná odpověď obsahuje jména uživatelů, kteří konkrétní polygon vytvořili, a jména kategorií.

```
$allPolygons = $pdo->prepare("select  
polygons.ID,polygons.vote,AsText(polygons.vertices) as vertices,users.nick,  
category.category from polygons join users on polygons.user = users.ID join  
category on polygons.categoryId = category.ID ");
```

Dále je vytvořen objekt `$dom` třídy `DOMDocument`. Tento objekt má několik metod, jejichž postupným voláním dosáhneme vytvoření správně strukturovaného KML souboru. Metoda `createElement()` vytvoří nový element, její první parametr určuje jméno elementu a druhý parametr obsah elementu. Metoda `appendChild()` připevní k rodičovskému elementu potomka, který je specifikovaný v parametru. Metoda `createAttribute()` vytvoří atribut elementu a její parametr je název atributu. Nejprve je vytvořena část KML dokumentu která se vyskytuje jen jednou a je nadřazená elementům, které se opakují pro každý polygon.

```
$dom = new DOMDocument('1.0','UTF-8');  
$node = $dom->createElementNS('http://www.opengis.net/kml/2.2','kml');  
$parNode = $dom->appendChild($node);  
$doc = $dom->createElement('Document');  
$node->appendChild($doc);
```

Dále je ve `foreach` cyklu vytvořena struktura stejná pro každý polygon a naplněna příslušnými daty. V elementu `description` jsou uloženy hodnoty ID polygonu, kategorie polygonu, jméno uživatele, který polygon vytvořil, a výsledný součet hlasů. V elementu `coordinates` jsou uloženy souřadnice každého bodu polygonu.

```
foreach ($plgResult as $polygon){
    // vytvoření kml souboru - viz kompletní zdrojový kód
```

Na konci skriptu je vytvořena hlavička, která určuje typ dokumentu a jeho kódování. Poslední řádek kódu volá metodu saveXML() objektu \$dom, která odešle zpět klientovi KML soubor.

Vytvoření souboru ke stažení probíhá obdobně, jako výše popsané generování KML souboru. Nejprve uživatel přejde do select módu a poté označí konkrétní polygon. Po kliknutí na polygon jsou do proměnné kmlDescription předána příslušná data metodou featureData a následně je spuštěna funkce KMLdataMode.

```
google.maps.event.addListener(kmlFromDB,"click",function(kmlEvent){
    kmlDescription = kmlEvent.featureData.description;
    KMLdataMode(kmlDescription);
```

Funkci KMLdataMode, která je umístěna v souboru layout.js, je předán parametr kmlDescription. Tato funkce nejprve naplní příslušný informační panel získanými daty o konkrétním polygonu a poté panel zobrazí v levé části mapy. Nakonec funkce zapíše ID polygonu do skrytého pole, které je využito pro identifikaci polygonu při odeslání požadavku na stáhnutí KML souboru.

```
function KMLdataMode(KMLData) {
    KMLdataArray = KMLData.split(",");
    // naplnění informačního panelu - viz kompletní zdrojový kód
    $(".KMLselect").hide(function() {
        $(".KLMvoting, .KMLdetails").show();});
    $("#polygonInput").slideUp("fast",function(){
        $("#KMLData").slideDown("fast");});
    $("#exportPolygonID").val(KMLdataArray[0]);}
```

Tlačítko pro stáhnutí polygonu potvrdí formulář se skrytým polem, kde je ID aktuálně označeného polygonu, a metodou post jej odešle na server, kde je zpracován v souboru exportXML.php. Soubor je generován obdobně jako je výše popsáno ve skriptu v genXML.php. S tím rozdílem, že databáze je dotazovaná pouze na jeden polygon.

```
$selectPolygons = $pdo->prepare("select
polygons.ID,polygons.vote,AsText(polygons.vertices) as vertices,users.nick,
category.category "
    . "from polygons join users on polygons.user = users.ID join
category on polygons.categoryId = category.ID "
    . "where polygons.ID = ?");
```

Po vytvoření struktury DOM objektu je do proměnné \$fileName uloženo ID stahovaného polygonu a aktuální datum. Poté je přidána hlavička se jménem souboru, která nutí klienta soubor stahovat, nikoliv zobrazovat.

```
$fileName = 'PolygonID'.$polygonResult['ID'] .strftime('_%d%m%Y.kml');
```



```
header('Content-Disposition: attachment;filename=' . $fileName);
header('Content-Type: text/xml; charset=utf-8');
echo $dom->saveXML();
```

### 3.2.5 Hlasování

V select módu je po kliknutí na některý polygon zobrazen informační panel v levé části mapy. V levé části tohoto panelu je zobrazen aktuální součet hlasů, které polygonu udělili uživatelé. Nad a pod tímto číslem jsou šipky, kterými uživatel může hlasovat. Tyto šipky volají funkci `voting()` v souboru `mashUpMap.js` a předávají funkci parametr s hodnotou `+1` nebo `-1`, podle toho, jak uživatel hlasoval. Funkce nejprve zkontroluje oprávnění uživatele, poté získá ID polygonu ze skrytého formuláře a následně odešle asynchronně požadavek na server metodou `post`. Podle dat vrácených serverem se funkce větví. Pokud hlasování proběhlo v pořádku, dojde k přičtení, respektive odečtení hlasu v informačním panelu a k asynchronnímu obnovení mapy. Nakonec je uživatel informován o započtení hlasu. Pokud hlasování selhalo z neznámého důvodu, je uživatel informován. Pokud uživatel pro tento polygon již hlasoval, je o tom informován a k hlasování nedojde.

```
function voting(voteValue) {
    if (checkStatus()){
        var polygonID = parseInt($("#KMLID").text());
        $.post('voting.php', {vote: voteValue, user: userID, polygon:
polygonID}, function(data){
            // zpracování dat ze serveru - viz kompletní zdrojový kód
```

Samotné hlasování na serveru provádí skript v souboru `voting.php`. Zde je nejprve sestaven dotaz na databázi, který vrací záznam konkrétního polygonu. Z tohoto záznamu je vyjmut atribut `voters`, který obsahuje ID všech uživatelů, kteří hlasovali pro tento konkrétní polygon. Všechna ID jsou v cyklu `foreach` porovnávána s ID uživatele, který právě hlasuje. Pokud je nalezena shoda, skript vrátí klientovi hodnotu, že uživatel již hlasoval a dojde k ukončení celého skriptu.

```
$query = $pdo->prepare("select * from polygons where ID= ?");
$query->execute(array($polygonID));
$polygon = $query->fetch();
$voters = explode(" ", $polygon["voters"]);
foreach ($voters as $voter){
    if ($voter == $user){
        echo 4; //already voted
        exit();
```

Když není nalezena shoda, skript pokračuje. Databázi je odeslán dotaz, který přičte, respektive odečte hlas a do atributu `voters` připojí ID hlasujícího uživatele. Pokud update proběhne v pořádku, vrátí skript klientovi potvrzující hodnotu `1`. Když skript selže, vrací hodnotu `2` nebo `3`, podle toho, v jakém bodě selhal.

---

```
$updateVoting = $pdo->prepare("update polygons set voters =
concat(voters,?),vote = vote+? where ID= ?");
$updateVoting->execute(array(" ".$user,$vote,$polygonID));
$updateVotingResult = $updateVoting->rowCount();
if ($updateVotingResult == 1){echo 1; //voting done
}else{echo 2; //voting failed}
```

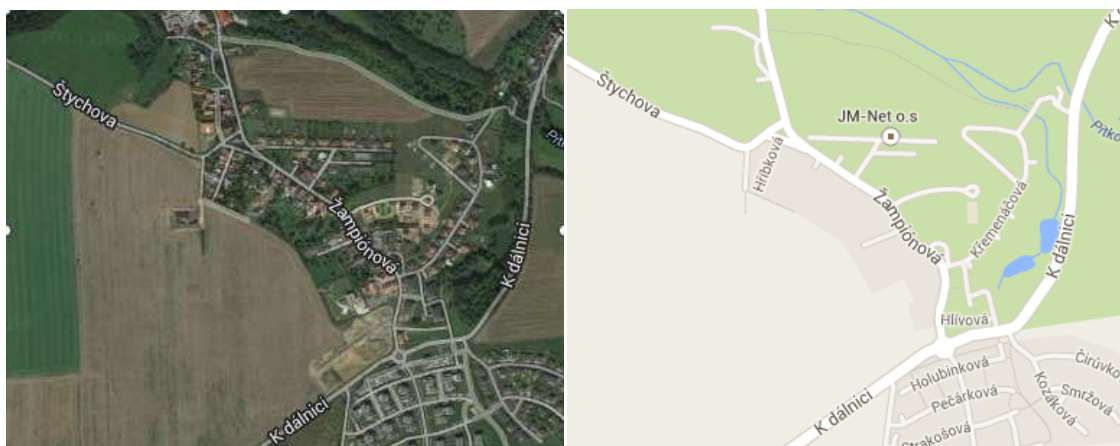
## 4 VÝSLEDKY A DISKUZE

V této kapitole je popsána výsledná aplikace a jsou zde shrnuty a diskutovány výsledky práce a dosažené cíle.

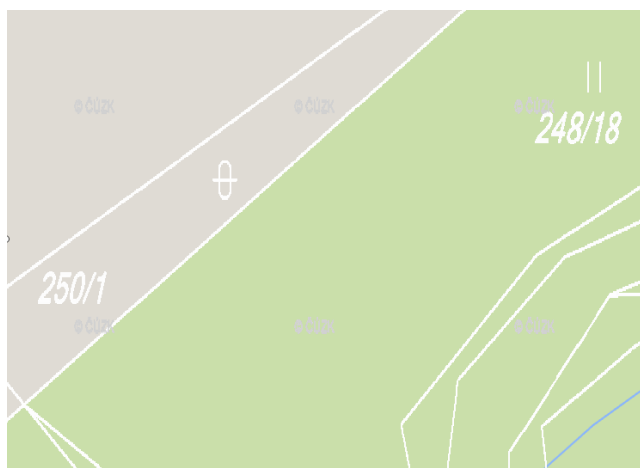
### 4.1 Popis vytvořené aplikace

Uživatelé jsou poskytována prostorová data ve formě mapy. Základní mapu lze přepínat mezi základní topografickou a hybridní satelitní mapou společnosti Google (obr. č. 4).

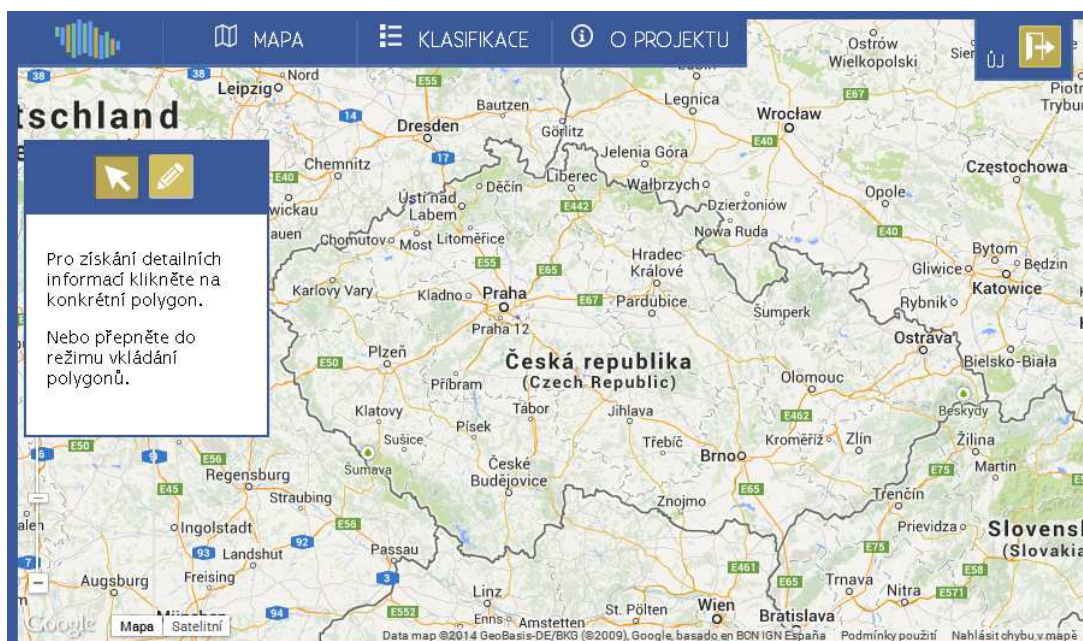
Obr. č. 4 Hybridní a základní mapa Google (zdroj: Google Maps, 2014)



Při přiblížení 17 a více se zobrazuje překryvná vrstva katastrální mapy se značkami druhů pozemků a jejich využití. Tato vrstva reprezentuje land use a land cover složku aplikace (obr. č. 5).

**Obr. č. 5 Vrstva katastrální mapy** (zdroj: Vlastní tvorba)

Aplikace je strukturována jako jedna hlavní stránka, na které probíhá veškerá interakce s uživatelem. V základním stavu je zobrazena základní mapa v takovém zvětšení, aby v náhledu byla přibližně celá Česká republika (náhled je závislý na velikosti okna prohlížeče). V hlavní navigační liště jsou tlačítka, která otevírají elementy s textovým obsahem. V elementu KLASIFIKACE je uživatel seznámen s částí značkového klíče katastrální mapy a může následně identifikovat chyby v katastrální mapě v porovnání s Google mapou nebo svou lokální znalostí. V elementu O PROJEKTU je seznámen s hlavním účelem aplikace. Dále je v základním náhledu v levém panelu uživatel vyzván k práci s aplikací (obr. č. 6).

**Obr. č. 6 Základní náhled aplikace přihlášeného uživatele** (zdroj: vlastní tvorba)

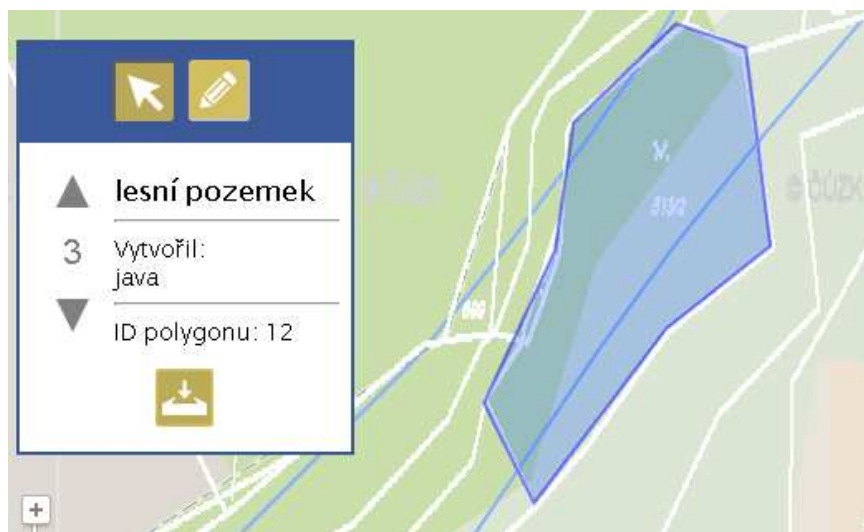
Pokud uživatel klikne na přepínací tlačítko s piktogramem tužky v levém ovládacím panelu, je aplikace přepnuta do módu zakreslování polygonů (draw mód). Uživatel následně může zakreslovat do mapy polygon krátkými kliknutími levého tlačítka myši. Dlouhým podržením levého tlačítka myši se

lze posouvat v mapě a kolečkem přibližovat a oddalovat. Na obrázku č. 7 je také vidět formulář typu select, ze kterého lze vybrat kategorii, o které se uživatel domnívá, že je v daném polygonu správná. Kdykoliv v průběhu zadávání polygonu lze tento proces přerušit kliknutím na tlačítko s piktogramem popelnice, otevřením panelů KLASIFIKACE a O PROJEKTU nebo přepnutím do módu select. Když uživatel dokončí polygon a vybere kategorii, může odeslat polygon tlačítkem s piktogramem šipky směřující vzhůru (obr. č. 7).

Obr. č. 7 Vkládání polygonu (zdroj: vlastní tvorba)



Po úspěšném odeslání se mapa obnoví a je zobrazen vložený polygon. Uživatel může zakreslit a odeslat další polygon, nebo přepnout do módu select tlačítkem s piktogramem kurzoru v horní části ovládacího panelu. V tomto módu lze kliknout na libovolný, dříve zadaný polygon a otevřít tím informační panel se základními informacemi o konkrétním polygonu. Informační panel je zobrazen na obrázku č. 8. Panel informuje uživatele o kategorii, kterou polygonu přiřadil jeho tvůrce, o jménu tvůrce a o identifikátoru polygonu ID. V levé části panelu je zobrazen součet hlasů všech hlasujících uživatelů a dvojice šipek. Šipkou vzhůru zvýší uživatel součet hlasů a vyjadřuje tím souhlas s polohovou a tematickou správností polygonu. Šipkou dolů je jeden hlas odečten a hlasující se vyjadřuje pro nesprávnost polygonu. Uživatel disponuje jen jedním hlasem pro každý polygon a v případě, že hlasuje znovu, je na tuto skutečnost upozorněn a hlas není započten.

**Obr. č. 8 Informační a hlasovací panel** (zdroj: vlastní tvorba)

Po kliknutí na tlačítko s piktogramem šipky směřující dolů začne stahování daného polygonu. V závislosti na individuálním nastavení klienta je soubor stažen do předem daného adresáře nebo je zobrazen dialog, kde uživatel vybere umístění. Tento soubor lze zobrazit v různých prohlížečkách KML souborů, z nichž některé podporují vyskakovací infoWindow, kde mohou být zobrazeny informace o polygonu. Cizí prohlížečky KML souborů nemusejí podporovat diakritiku. V příloze č. 6 je struktura KML souborů, které uživatelé mohou stahovat. Jméno staženého souboru je složeno z ID konkrétního polygonu a z data dne stažení.

Uživatelé se mohou bezprostředně po registraci přihlásit do aplikace, prohlížet detaily a stahovat polygony. Pro vkládání polygonů a hlasování musí být aktivován uživatelský účet. Aktivaci uživatel dokončí otevřením odkazu, který mu byl zaslán mailem po registraci.

Pro správné zobrazení aplikace je nutné, aby v prohlížeči byl povolen JavaScript. Aplikace byla vyvíjena a je otestována pro klienty Google Chrome a Mozilla Firefox a je umístěna na stránkách <http://bp.test.cernik.name>.

## 4.2 Diskuse

Hlavním cílem této práce bylo vytvořit webovou aplikaci pro sběr a validaci prostorových land use a land cover dat. Ačkoliv v průběhu vývoje samotné aplikace se autor střetával s mnoha problémy a nejednou došel do situace, kdy musel znatelně revidovat svoje současné znalosti v oboru informatiky, hlavní cíl byl naplněn. Výsledná aplikace je ve svých hlavních bodech funkční. Přesto se funkcionalita v některých situacích stává problémovou. Jde zejména o situaci, kdy uživatel zakreslí polygon uvnitř stávajícího polygonu. Nový polygon se tak stává uživatelům nepřístupný a při kliknutí se označí polygon vnější. Proto nelze vnitřní polygon označit, stahovat a hlasovat pro něj. Dalším hypotetickým problémem je funkčnost aplikace, pokud by byla překonána limitní velikost KML



souboru, který přenáší polygony z databáze do klienta. Podle dokumentace Google Maps API je současný datový limit KML souborů 10 MB. To znamená, že v současném stavu je aplikace schopna zobrazovat několik tisíc polygonů, podle jejich složitosti. Aktuálně je tento limit daleko od nároků aplikace, ale jak bude růst počet uživatelů a počet polygonů, bude tento problém třeba řešit. Také je pravděpodobné, že až tento soubor naroste do velikosti několika MB, bude aplikace výrazně zpomalena.

Aktuálně stav aplikace umožňuje stahovat polygony po jednom v souborech KML. Budoucí vývoj a ohlasy uživatelů ukáží, zda bude potřeba tuto funkci rozšířit, například o stahování většího množství vybraných polygonů nebo všech polygonů.

Dalším cílem práce bylo popsat a použít v aplikaci některé land use a land cover zdroje. Výsledky tohoto cíle jsou značně rozporupné. Z rešeršovaných zdrojů bylo možné použít pouze jeden, katastrální mapy. Při návrhu aplikace bylo zamýšleno připojovat tyto zdroje do mapy pomocí WMS ale tento způsob distribuce je značně omezený. V době psaní této práce byly ze čtyř rešeršovaných datových zdrojů distribuovány pouze dva formou WMS. Byly to katastrální mapy a Urban Atlas. Nicméně připojení WMS do aplikace pomocí Google Maps API se ukázalo jako poměrně komplikované a aktuální vývojářské schopnosti autora nejsou natolik vysoké, aby zvládl implementaci. Připojení katastrálních map bylo odvozeno ze zdrojového kódu aplikace Katastr nemovitostí 2.0.

V souladu s moderními přístupy označované jako web 2.0 umožňuje aplikace uživatelům vkládat vlastní obsah ve formě libovolných polygonů. Tento neřízený proces je třeba nějakým způsobem validovat. Autor se rozhodl pro použití modifikovaného reputačního systému. Všichni uživatelé mohou hlasovat o správnosti jakéhokoliv polygonu a vyjadřovat svůj souhlas nebo nesouhlas s polohovou a tematickou přesností polygonů. Tento přístup usnadňuje práci administrátorovi nebo editorovi, který tak je informován o relativní validnosti dat a na základě hlasování může nesprávné polygony mazat. Konkurenční přístup validace vkládaných dat používá projekt Geo-wiki. Zde uživatelé dokládají validnost vkládáním fotografií dotčeného území a na konkrétní oblast jsou opakovaně dotazováni různí uživatelé. Oba přístupy validace vyžadují aktivní přístup editorů.

Dalším dílčím cílem bylo uvést čtenáře do problematiky dynamického webu. V rámci rozsahu této práce byly zevrubně popsány vybrané technologie, které se pro dynamický web používají a zároveň byly použity pro výslednou mapovou aplikaci. Byly popsány a použity technologie serverového (PHP) a klientského skriptování (JavaScript) a vybraný databázový systém (MySQL). Také byly popsány vybrané API, které jsou velmi nápomocné při vyvíjení mapové on-line aplikace. Použitá mapová API společnosti Google obsahuje množství tříd a procedur, které významně ulehčují vývoj. Vzhledem k objemnosti těchto témat nelze tento text považovat za příručku popsaných technologií, ale může pomoci začínajícímu webovému vývojáři zorientovat se v tématu. Zásadním

dilematem bylo rozhodnutí zahrnout do textu práce podkapitolu 3.1, která pojednává o návrhu aplikace. Nakonec se autor rozhodl tuto podkapitolu ponechat v textu, protože návrh aplikace považuje za zcela zásadní část celého procesu vývoje a každý vývojář, zvláště začínající, by neměl toto téma podceňovat.



## 5 ZÁVĚR

Tato práce je věnována vývoji on-line mapové aplikace. Výsledná aplikace propojuje moderní technologie dynamického webu a aktuální téma land use a land cover změn. Výsledná aplikace splňuje základní funkcionalitu, jak byla navržena. Nejprve byl vytvořen komplexní návrh aplikace po stránce logiky, grafiky, bezpečnosti aplikace a datového úložiště. Land use a land cover data reprezentuje katastrální mapa, která byla do aplikace připojena pomocí standardu WMS. Aplikace umožňuje uživatelům vkládat polygony a zaznamenat tak chyby v katastrální mapě. O správnosti vložených polygonů mohou následně hlasovat všichni uživatelé. Uživatelům je také umožněno stahovat jednotlivé polygony ve formátu KML. Jako užitečné se ukázalo použití aplikačního programovacího rozhraní Google Maps, které významně urychlilo a usnadnilo vývoj.

Problematickým se ukázalo použití WMS při získávání land use a land cover dat. Protože použité API nepodporuje WMS přímo, je nutné tuto vrstvu simulovat překryvným obrazem a implementace je poměrně složitá. Vzhledem k omezeným programátorským zkušenostem autora nabylo možné připojit do aplikace data projektu Urban Atlas. Druhým problémem WMS je neexistence této služby na území ČR pro data projektů Corine land cover a GlobCover. Alternativy k WMS jsou následující. Lze získat binární soubory Corine land cover a GlobCover a následně je poskytovat z webového serveru. Tento přístup autor netestoval a pravděpodobně by byl problematický velký objem dat, přenášený ze serveru klientovi. Další alternativou může být vytvoření vlastního mapového serveru, který by poskytoval WMS zmiňovaných datových souborů. Tato služba by poskytla prostor pro vývoj mnoha aplikací a byla by jistě přínosná pro další výzkum land cover a land use procesů.

Potenciální problém aplikace spočívá v omezení velikosti KML souboru, který zobrazuje polygony v mapě. Řešením se zdá být použití jiného ukládání dat a na místo databázového systému MySQL použít technologii Google Fusion Tables API. Další alternativou je získávat z databáze a vykreslovat jen ty polygony, které jsou v aktuálním náhledu. Oba přístupy jsou pro autora velkou výzvou a bude se jim dále věnovat.

Podobně jako studie týkající se projektu Geo-wiki, autor tohoto textu vidí problém crowdsourcingu při získávání geografických dat v počtu aktivních uživatelů. Zvláště použitý validační

systém, který je založen na reputačních bodech jednotlivých polygonů, bude v reálném provozu aplikace velice náchylný na počet aktivních uživatelů. Aby se prokázala validita jednotlivých polygonů na základě hlasování, pravděpodobně by byly potřeba desítky hlasů pro každý polygon. Tato otázka ovšem zůstává otevřena a je zde prostor pro další výzkum. Zároveň další vývoj aplikace může být nápomocný pro získávání nových uživatelů. Zde je prostor pro návrh a vývoj mobilní aplikace nebo rozšíření reputačního systému z polygonů na uživatele, který by na jednu stranu uživatele motivoval v používání a na druhou stranu by podpořil validnost dat.

Pokud bude aplikace úspěšná, získá více uživatelů a bude pracovat s velkým objemem dat, je nezbytné vyvinout administrační rozhraní pro správce aplikace, které by usnadnilo spravování dat a administraci uživatelských účtů. V současném stavu je pro tyto úkony potřeba vstupovat přímo do databáze a pracovat s uživatelsky nepřívětivým rozhraním databázového systému. Autor si dává za cíl toto administrační rozhraní vytvořit.

Dílním cílem bylo vytvořit text, který by byl nápomocen začínajícím vývojářům zorientovat se v tématu. Tohoto cíle bylo dosaženo strukturou a obsahem textu. V kontextu s tímto cílem považuje autor za zvláště přínosnou podkapitolu 3.1 návrh aplikace. Pro úplnost je třeba dodat, že oblast webového vývoje je velice rozsáhlé a rychle se měnící téma a to co bylo relevantní před rokem, může být dnes zcela jinak. Autor před započítím tohoto projektu neměl téměř žádné znalosti a zkušenosti s programováním pro web a v průběhu této práce mnohokrát revidoval svoje znalosti, což vedlo k nespočetnému předělávání aplikace. Proto autor doporučuje vyhledávat vždy ty nejaktuálnější informace, co se týká webového vývoje.

Výsledná aplikace je dostupná na URL adrese <http://bp.test.cernik.name>.

## POUŽITÉ ZDROJE

- ALBORS, J., RAMOS, J. C., HERVAS, J. L. (2008): *New learning network paradigms: Communities of objectives, crowdsourcing, wikis and open source*. International Journal of Information Management. 2008, Vol. 28, Issue 3, p. 194 – 202
- ARDIELLI, J. a spol. (2009): *Modern tools for development of interactive web map applications for visualization spatial data on the internet*. Acta Montanistica Slovaca. 2009, Vol. 14, Issue. 1, p. 4-11.
- BÜTTNER, G. et al. (2002): Corine land cover update 2000: *Technical guidelines*. European environment agency. Kodaň: European environment agency, 2002. 56 p. ISBN: 92-9167-511-3
- COPERNICUS: The European Earth Observation Programme (2013): *Corine Land Cover* [on-line]. Poslední revize 9. 12. 2013. Dostupné z URL: <http://land.copernicus.eu/pan-european/corine-land-cover> (cit. 7. 2. 2014)
- COPERNICUS: The European Earth Observation Programme (2011): *Urban Atlas mapping guide* [on-line]. Dostupné z URL: <http://www.eea.europa.eu/data-and-maps/data/urban-atlas#tab-methodology> (cit. 7. 2. 2014)
- ČESKÁ REPUBLIKA (2013): *Vyhláška č. 357 ze dne 1. listopadu 2013 Zákon o katastru nemovitostí*. In Sbírka zákonů České republiky. 2013. Částka 141/2013
- ČÚZK: Český úřad zeměměřický a katastrální (2013): *Zpřesnění geometrického a polohového určení pozemků* [on-line]. Poslední revize 31. 1. 2013. Dostupné z URL: <http://www.cuzk.cz/Katastr-nemovitosti/Digitalizace-a-vedeni-katastralnich-map/Zpresneni-geometrickeho-a-polohoveho-urceni-pozemk.aspx> (cit. 7. 2. 2014)

- ECMA (2014): *ECMAScript Documentation* [on-line]. Poslední revize květen 2014, Dostupné z URL: <http://www.ecmascript.org/docs.php> (cit. 12. 5. 2014)
- EEA: European environment agency (1995): *CORINE land cover* [on-line]. Dostupné z URL: <http://www.eea.europa.eu/publications/COR0-landcover> (staženo 2. 2. 2014)
- EEA: European environment agency (2010): *Urban Atlas* [on-line]. Poslední revize 17. 4. 2013. Dostupné z URL: <http://www.eea.europa.eu/data-and-maps/data/urban-atlas> (cit. 7. 2. 2014)
- EEA: European environment agency (2013): *Corine Land Cover 2006 seamless vector data* [on-line]. Poslední revize prosinec 2013. Dostupné z URL: <http://www.eea.europa.eu/data-and-maps/data/clc-2006-vector-data-version-3#tab-additional-information> (staženo 13. 5. 2014)
- ESA: European Space Agency (2008): *GLOBCOVER Products Description and Validation Report* [on-line]. Dostupné z URL: <http://due.esrin.esa.int/globcover/> (staženo 2. 2. 2014)
- ESA: European Space Agency (2011): *GLOBCOVER 2009-Products Description and Validation Report* [on-line]. Dostupné z URL: <http://due.esrin.esa.int/globcover/> (staženo 2. 2. 2014)
- ESRI (2014): *ArcGIS API for JavaScript* [on-line]. Poslední revize 6. 1. 2014. Dostupné z URL: <https://developers.arcgis.com/javascript/> (cit. 8. 2. 2014)
- FRITZ, S. a spol (2009): *Geo-Wiki.Org: The Use of Crowdsourcing to Improve Global Land Cover*. Remote Sensing. 2009, roč. 1, č. 3, s. 345-354
- GEO-WIKI (2014): *Geo-wiki: Data source* [on-line]. Dostupné z URL: <http://www.geo-wiki.org/data-source/> (cit 9. 2. 2014)
- GILES, J. (2006): *Wikipedia rival calls in the experts*. Nature. 2006, Vol. 443, Issue 7111, p. 493-493
- GOOGLE (2013): *Google Maps JavaScript API v3* [on-line]. Poslední revize 15.12.2013. Dostupné z URL: <https://developers.google.com/maps/documentation/javascript/tutorial> (cit. 9. 2. 2014)

- GOODCHILD, M. F., LI, L. (2012): *Assuring the quality of volunteered geographic information*. Spatial Statistics. 2012, Vol. 1, p. 110 – 120
- HAKLAY, M. (2010): *How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets*. Environment and Planning B: Planning and Design. 2010, Vol. 37, Issue 4, p. 682 – 703
- HERRING, J. (2011): *OpenGIS® Implementation Standard for Geographic information: Simple feature access - Part 1: Common architecture* [on-line], Publikováno 28. 5. 2011, Dostupné z URL <http://www.opengeospatial.org/standards/sfa> (staženo 12. 5. 2014)
- HOWE, J. (2006): *The rise of crowdsourcing*. Wired. 2006, Vol. 14
- JQUERY (2014): *The jQuery Foundation* [on-line]. Dostupné z URL: <https://jquery.org/> (cit. 12. 5. 2014)
- ORACLE (2014): *MySQL Documentation: Reference Manuals* [on-line]. Dostupné z URL: <http://dev.mysql.com/doc/> (cit. 12. 5. 2014)
- PHP: Hypertext Preprocessor (2014): *The PDOStatement class* [on-line]. Dostupné z URL: <http://cz2.php.net/manual/en/class.pdostatement.php> (cit. 12. 5. 2014)
- SEZNAM (2014): *Mapy API verze 4.8 – Hanzelka a Zikmund* [on-line]. Dostupné z URL: <http://api4.mapy.cz/> (cit. 7. 2. 2014)
- WILSON, T. (2008): *OGC® KML* [on-line], Publikováno 14. 4. 2008, Dostupné z URL: <http://www.opengeospatial.org/standards/kml> (staženo 12. 5. 2014)
- ZBIEJCZUK, A. (2007): *Web 2.0 – charakteristika a služby*. Brno, 2007. Diplomová práce. Masarykova Univerzita v Brně, Fakulta sociálních studií, Katedra mediálních studií a žurnalistiky. Dostupné z URL: [http://www.zbiejczuk.com/adam/zbiejczuk\\_web20.pdf](http://www.zbiejczuk.com/adam/zbiejczuk_web20.pdf)

## SEZNAM PŘÍLOH

Příloha 1: Globální klasifikační legenda GlobCover 2009

Příloha 2: Nomenklatura Corine Land Cover

Příloha 3: Nomenklatura Urban Atlas

Příloha 4: Katastrální mapa: Druh pozemku

Příloha 5: Katastrální mapa: Způsob využití pozemku























Příloha 6: Struktura KML souboru ke stažení

Příloha 7: Přiložené CD-R:

- Složka BP (vlastní text práce)

- Složka Aplikace (veškeré soubory aplikace)

**Příloha 1: Globální klasifikační legenda GlobCover 2009** (zdroj: ESA, 2008)

Value	GlobCover global legend	
11	Post-flooding or irrigated croplands	
14	Rainfed croplands	
20	Mosaic Cropland (50-70%) / Vegetation (grassland, shrubland, forest) (20-50%)	
30	Mosaic Vegetation (grassland, shrubland, forest) (50-70%) / Cropland (20-50%)	
40	Closed to open (>15%) broadleaved evergreen and/or semi-deciduous forest (>5m)	
50	Closed (>40%) broadleaved deciduous forest (>5m)	
60	Open (15-40%) broadleaved deciduous forest (>5m)	
70	Closed (>40%) needleleaved evergreen forest (>5m)	
90	Open (15-40%) needleleaved deciduous or evergreen forest (>5m)	
100	Closed to open (>15%) mixed broadleaved and needleleaved forest (>5m)	
110	Mosaic Forest/Shrubland (50-70%) / Grassland (20-50%)	
120	Mosaic Grassland (50-70%) / Forest/Shrubland (20-50%)	
130	Closed to open (>15%) shrubland (<5m)	
140	Closed to open (>15%) grassland	
150	Sparse (>15%) vegetation (woody vegetation, shrubs, grassland)	
160	Closed (>40%) broadleaved forest regularly flooded - Fresh water	
170	Closed (>40%) broadleaved semi-deciduous and/or evergreen forest regularly flooded - Saline water	
180	Closed to open (>15%) vegetation (grassland, shrubland, woody vegetation) on regularly flooded or waterlogged soil - Fresh, brackish or saline water	
190	Artificial surfaces and associated areas (urban areas >50%)	
200	Bare areas	
210	Water bodies	
220	Permanent snow and ice	

**Príloha 2: Nomenklatura Corine Land Cover** (zdroj: EEA, 2013)

CODE	LABEL2	LABEL3
	<b>Artificial surfaces</b>	
111	Urban fabric	Continuous urban fabric
112	Urban fabric	Discontinuous urban fabric
121	Industrial, commercial and transport units	Industrial or commercial units
122	Industrial, commercial and transport units	Road and rail networks and associated land
123	Industrial, commercial and transport units	Port areas
124	Industrial, commercial and transport units	Airports
131	Mine, dump and construction sites	Mineral extraction sites
132	Mine, dump and construction sites	Dump sites
133	Mine, dump and construction sites	Construction sites
141	Artificial, non-agricultural vegetated areas	Green urban areas
142	Artificial, non-agricultural vegetated areas	Sport and leisure facilities
211	Arable land	Non-irrigated arable land
212	Arable land	Permanently irrigated land
213	Arable land	Rice fields
221	Permanent crops	Vineyards
222	Permanent crops	Fruit trees and berry plantations
223	Permanent crops	Olive groves
231	Pastures	Pastures
241	Heterogeneous agricultural areas	Annual crops associated with permanent crops
242	Heterogeneous agricultural areas	Complex cultivation patterns
243	Heterogeneous agricultural areas	Land principally occupied by agriculture, with significant areas of natural vegetation
244	Heterogeneous agricultural areas	Agro-forestry areas
	<b>Forest and semi natural areas</b>	
311	Forests	Broad-leaved forest
312	Forests	Coniferous forest
313	Forests	Mixed forest
321	Scrub and/or herbaceous vegetation associations	Natural grasslands
322	Scrub and/or herbaceous vegetation associations	Moors and heathland
323	Scrub and/or herbaceous vegetation associations	Sclerophyllous vegetation
324	Scrub and/or herbaceous vegetation associations	Transitional woodland-shrub
331	Open spaces with little or no vegetation	Beaches, dunes, sands
332	Open spaces with little or no vegetation	Bare rocks
333	Open spaces with little or no vegetation	Sparsely vegetated areas
334	Open spaces with little or no vegetation	Burnt areas
335	Open spaces with little or no vegetation	Glaciers and perpetual snow
	<b>Wetlands</b>	
411	Inland wetlands	Inland marshes



CODE	LABEL2	LABEL3
412	Inland wetlands	Peat bogs
421	Maritime wetlands	Salt marshes
422	Maritime wetlands	Salines
423	Maritime wetlands	Intertidal flats
	<b>Water bodies</b>	
511	Inland waters	Water courses
512	Inland waters	Water bodies
521	Marine waters	Coastal lagoons
522	Marine waters	Estuaries
523	Marine waters	Sea and ocean

Tučně jsou vyznačené třídy nejvyšší úrovně.

**Příloha 3: Nomenklatura Urban Atlas** (zdroj: EEA, 2013)

Urban Atlas No.	Vector Data Code	Nomenclature	Additional Information
	GSELUA_yy		
1		Artificial surfaces	
1.1		Urban Fabric	
<b>1.1.1</b>	<b>11100</b>	<b>Continuous Urban Fabric (S.L. &gt; 80%)</b>	FTS <sup>1</sup> required
1.1.2	11200	Discontinuous Urban Fabric (S.L. 10% - 80%)	
<b>1.1.2.1</b>	<b>11210</b>	<b>Discontinuous Dense Urban Fabric (S.L. 50% - 80%)</b>	FTS required
<b>1.1.2.2</b>	<b>11220</b>	<b>Discontinuous Medium Density Urban Fabric (S.L. 30% - 50%)</b>	FTS required
<b>1.1.2.3</b>	<b>11230</b>	<b>Discontinuous Low Density Urban</b>	FTS required
<b>1.1.2.3</b>	<b>11230</b>	<b>Discontinuous Low Density Urban Fabric (S.L. 10% - 30%)</b>	FTS required
<b>1.1.2.4</b>	<b>11240</b>	<b>Discontinuous Very Low Density Urban Fabric (S.L. &lt; 10%)</b>	FTS required
1.1.3	11300	Isolated structures	
1.2		Industrial, commercial, public, military, private and transport units	
<b>1.2.1</b>	<b>12100</b>	<b>Industrial, commercial, public, military and private units</b>	zoning data / field check recommended
1.2.2	12200	Road and rail network and associated land	COTS <sup>2</sup> navigation data required
<b>1.2.2.1</b>	<b>12210</b>	<b>Fast transit roads and associated land</b>	COTS navigation data required

**Příloha 3 pokračování: Nomenklatura Urban Atlas (zdroj: EEA, 2013)**

Urban Atlas No.	Vector Data Code	Nomenclature	Additional Information
1.2.2.2	12220	Other roads and associated land	COTS navigation data required
1.2.2.3	12230	Railways and associated land	COTS navigation data required
1.2.3	12300	Port areas	zoning data / field check recommended
1.2.4	12400	Airports	zoning data / field check recommended
1.3		Mine, dump and construction sites	
1.3.1	13100	Mineral extraction and dump sites	
1.3.3	13300	Construction sites	
1.3.4	13400	Land without current use	
1.4		Artificial non-agricultural vegetated areas	
1.4.1	14100	Green urban areas	
1.4.2	14200	Sports and leisure facilities	
2	20000	Agricultural areas, semi-natural areas and wetlands	1 ha MMU
3	30000	Forests	1 ha MMU
5	50000	Water	1 ha MMU

**Příloha 4: Katastrální mapa: Druhy pozemku** (zdroj: ČESKÁ REPUBLIKA, 2013)

Kód	Název	Zkráceně	Charakteristika druhu pozemku pro účely katastru
2	orná půda		Pozemek orné půdy podle zákona o zemědělství.
3	chmelnice		Pozemek chmelnice podle zákona o zemědělství.
4	vinice		Pozemek vinice podle zákona o zemědělství.
5	zahrada		Pozemek, a) na němž se trvale a převážně pěstuje zelenina, květiny a jiné zahradní plodiny, zpravidla pro vlastní potřebu, b) souvisle osázený ovocnými stromy nebo ovocnými keři, který zpravidla tvoří souvislý celek s obytnými a hospodářskými budovami.
6	ovocný sad	ovoc. sad	Pozemek ovocného sadu podle zákona o zemědělství.
7	trvalý travní porost	travní p.	Pozemek trvalého travního porostu podle zákona o zemědělství.
10	lesní pozemek	lesní poz.	Pozemek s lesním porostem a pozemek, u něhož byly lesní porosty odstraněny za účelem jejich obnovy, lesní průsek a nezpevněná lesní cesta, není-li širší než 4 m, a pozemek, na němž byly lesní porosty dočasně odstraněny na základě rozhodnutí orgánu státní správy lesů [§ 3 odst. 1 písm. a) zákona č. 289/1995 Sb.].
11	vodní plocha	vodní pl.	Pozemek, na němž je koryto vodního toku, vodní nádrž, močál, mokřad nebo bažina.
13	zastavěná plocha a nádvoří	zast. pl.	Pozemek, na němž je a) budova včetně nádvoří (tj. části zastavěného stavebního pozemku, obsahující dvůr, vjezd, drobné stavby, bazén, zatravněné plochy, okrasné záhony a jiné přiléhající plochy, které slouží k lepšímu užívání stavby), vyjma skleníku, který je v katastru evidován jako budova, postaveného na zemědělském nebo lesním pozemku, budovy postavené na lesním pozemku a budovy evidované na pozemku vodní plocha, b) společný dvůr, c) zbořeniště, d) vodní dílo.
14	ostatní plocha	ostat. pl.	Pozemek neuvedený v předcházejících druzích pozemků.

**Příloha 5: Katastrální mapa: Způsob využití pozemku** (zdroj: ČESKÁ REPUBLIKA, 2013)

Kód	Název	Zkráceně	Význam	Kód druhu p.
1	skleník, pařeniště	skleník-pařeniš.	Na pozemku je zřízen skleník nebo pařeniště.	2 až 7 a 10
2	školka		Na pozemku je zřízena školka ovocných, lesních nebo okrasných stromů, viničná školka nebo školka pro chmelovou sáď.	
3	plantáž dřevin		Na pozemku je semenná plantáž, plantáž energetických dřevin, vánočních stromků, porost určitého dřevinného druhu, zpravidla monokultura, s intenzivním agrotechnickým obděláváním půdy pro dosažení rychlé a vysoké produkce dřevní hmoty apod.	2, 7 až 10 a 14
4	les jiný než hospodářský	les(ne hospodář)	Pozemek zařazený do kategorie lesy ochranné a lesy zvláštního určení podle § 7 a 8 zákona č. 289/1995 Sb.	10
5	lesní pozemek, na kterém je budova	les s budovou	Lesní pozemek, na kterém je budova, ale pozemek není odňat plnění funkcí lesa.	
6	rybník		Umělá vodní nádrž určená především k chovu ryb s možností úplného a pravidelného vypouštění.	11
7	koryto vodního toku přirozené nebo upravené	tok přirozený	Koryto vodního toku, které vzniklo působením tekoucí vody a dalších přírodních faktorů (bystřina, potok, řeka) nebo jehož přírodní charakter je změněn technickými zásahy (např. břehovým opevněním) nebo ohrazováním.	11
8	koryto vodního toku umělé	tok umělý	Koryto vodního toku, které bylo vytvořeno uměle (například opevněné koryto vodního toku, průplav, kanál apod.).	
9	vodní nádrž přírodní	nádrž přírodní	Pozemek, na kterém je vodní nádrž, která nebyla vytvořena záměrnou lidskou činností (například jezero, přírodní deprese naplněná vodou apod.).	
10	vodní nádrž umělá	nádrž umělá	Pozemek, na kterém je vodní nádrž vytvořená záměrnou lidskou činností s výjimkou rybníku a bazénu ke koupání (například velká vodní nádrž vytvořená přehradou, malá vodní nádrž, nádrž vytvořená zatopením vytěžených ploch apod.).	
11	zamokřená plocha	zamokřená pl.	Zemský povrch trvale nebo po převážnou část roku rozbředlý (močál, mokřad, bažina).	11 a 14
12	společný dvůr		Společný dvůr v podílovém spoluvlastnictví vlastníků bytových domů, které s tímto pozemkem sousedí.	

Kód	Název	Zkráceně	Význam	Kód druhu p.
13	zbořeniště		Pozemek se zbořenou budovou nebo zříceninou.	13
14	dráha		Pozemek, na kterém je dráha železniční, tramvajová, trolejbusová nebo lanová (§ 1 zákona č. 266/1994 Sb.) s vlastní dopravní cestou.	14
15	dálnice		Pozemek, na kterém je dálnice a její součásti (§ 4 zákona č. 13/1997 Sb.).	
16	silnice		Pozemek, na kterém je silnice I. až III. třídy a její součásti (§ 5 zákona č. 13/1997 Sb.).	
17	ostatní komunikace	ostat, komunikace	Pozemek, na kterém je místní nebo účelová komunikace (včetně zpevněné lesní komunikace) a její součásti (§ 6 a 7 zákona č. 13/1997 Sb.).	10 a 14
18	ostatní dopravní plocha	ost. dopravní pl.	Letiště, přístav, veřejné parkoviště (pokud není součástí pozemní komunikace).	14
19	zeleň		Okrasná zahrada, uliční a sídlištní zeleň, park a jiná plocha funkční a rekreační zeleně.	
20	sportoviště a rekreační plocha	sport. arekr. pl.	Hřiště, stadion, koupaliště, sportovní dráha a jízdárna, střelnice, autokemp, tábořiště apod.	10 a 14
21	hřbitov, urnový háj	hřbitov-urn. háj	Hřbitov, urnový háj.	14
22	kulturní a osvětová plocha	kult. a osvět. pl.	Botanická a zoologická zahrada, skanzen, amfiteátr, památník apod.	
23	manipulační plocha	manipulační pl.	Manipulační a skladová plocha.	
24	dobývací prostor	dobývací prost.	Prostor jednoho nebo více výhradních ložisek nebo prostor jen části výhradního ložiska.	2 až 10 a 14
25	skládka		Skládka odpadu.	14
26	jiná plocha		Pozemek nevyužívaný žádným z ostatních vyjmenovaných způsobů včetně pozemku, na kterém je postavena stavba, která se v katastru neeviduje.	
27	neplodná půda		Pozemek, na němž se nachází prudký svah, skála a jiné neplodné půdy, kterými se rozumí zejména zarostlé rokle, vysoké meze s křovinami nebo s kamením, kamenitý terén, ochranné hráze, bermy u regulovaných vodních toků a pozemek, který neposkytuje trvalý užitek z jiných důvodů, zejména plocha zarostlá křovinami nebo zanesená šterkem nebo kamením, s výjimkou případů, kdy stav pozemku je důsledkem lidské činnosti nebo nečinnosti a pozemek lze opět uvést do stavu umožňujícího jeho trvalé hospodářské využití.	

Kód	Název	Zkráceně	Význam	Kód druhu p.
28	vodní plocha, na které je budova	vod. pl. s budovou	Pozemek vodní plochy, na kterém je postavena budova.	11
29	fotovoltaická elektrárna	foto. elektrárna	Fotovoltaická elektrárna.	2 až 14

**Příloha 6: Struktura KML souboru ke stažení (zdroj: vlastní tvorba)**

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark ID="12">
      <name>
        Plocha kategorie: lesní pozemek
      </name>
      <description>
        <![CDATA[
          <ul>
            <li> ID polygonu: 12</li>
            <li> Vytvořil: java</li>
            <li> Výsledek hlasování: 3</li>
          </ul>
        ]]>
      </description>
      <MultiGeometry>
        <Polygon>
          <outerBoundaryIs>
            <LinearRing>
              <coordinates>
                14.553533,50.009285,0.0
                14.553747,50.009466,0.0
                14.553916,50.009552,0.0
                14.553876,50.009758,0.0
                14.553763,50.009783,0.0
                14.553597,50.009680,0.0
                14.553567,50.009547,0.0
                14.553452,50.009389,0.0
                14.553533,50.009285,0.0
              </coordinates>
            </LinearRing>
          </outerBoundaryIs>
        </Polygon>
      </MultiGeometry>
    </Placemark>
  </Document>
</kml>
```